Facial Expression Recognition Based on 3D Face Reconstruction

A dissertation presented for the degree of

Doctor of Engineering

Theekapun Charoenpong

Advisor: Hiroyuki Hase

Department of System Design Engineering

Graduate School of Engineering

University of Fukui

Japan

September, 2008

CONTENTS

Abstract

Due to a problem of current research occurring when recognizing facial expressions from a 2.5D partial face data set taken from any viewpoint ranging from -45 degrees to +45 degrees, we propose a new effective method for recognizing facial expressions based on 3D face reconstruction. This proposed method consists of two parts. In the first part, a 3D face is reconstructed by using an ellipse fitting technique. A face vector is used to detect a symmetry plane. By using the symmetry plane and real face data, the 3D face can be reconstructed. In the second part, facial expression is represented in terms of the change of crossing points on a face plane. Two crossing point analysis methods consisting of a crossing point distribution method and a displacement vector method are used to analyze distribution of crossing points. By using a support vector machines method, facial expressions can be recognized. Four facial expressions which were used in experiments were neutral, anger, surprise and smiling. The results show the feasibility of the proposed method.

In the chapter 2, a 3D face is reconstructed. The data from each cross section of the head is fitted by an ellipse fitting technique. In each cross section of the head data, a vector pointing from a center of the ellipse to the facial surface is defined as a semi-major axis vector. Based on the experiments of all viewpoints, each cross section shows the different average ellipse fitting error. According to this property, the semi-major axis vector under the condition that the average ellipse fitting error of its cross section is less than 1 mm tend to point to the surface near the nose. By averaging the semi-major axis vectors obtained from one viewpoint, a face vector can be calculated. A nose ridge detection algorithm is used to detect the nose tip and the nose ridge. Vectors which are the projection of the face vector in the cross sections in the nose region are defined as projected face vector. Points on the facial surface in each cross section are projected onto the projected face vector. The projected points from the nose tip and nose ridge always show farthest distance from the center of the ellipse. To detect the nose tip and the nose ridge, the projected face vector is corrected to the point which shown the farthest distance. By using the nose tip and the nose ridge points, two vectors are obtained. One is a nose ridge vector. This vector is computed by a Principle Components Analysis method (PCA). Another one is a nose vector. The nose

vector points from the center of the head to the average of the nose ridge point. By using the two vectors, a symmetry plane is defined. This symmetry plane is corrected by using a symmetry plane correction algorithm. The 3D face is then reconstructed.

In the chapter 3, a facial expression is represented in a face plane. From a reconstructed 3D face, facial surface is extracted by using a L*a*b* color space. A face plane algorithm is used to represent a facial expression in terms of crossing points. A face plane is a virtual plane across the head. Four neighboring points, which are points on the facial surface, are used to compute a normalized vector. The normalized vectors point to the center of the head. By using the normalized vectors, the face plane is computed. A point on the face plane which a normalized vector points through is defined as crossing point. In each facial expression, the character of the crossing point distribution is different. The crossing point distribution of an expression face is compared with a neutral face from the same person for recognition. Two methods are used to analyse the crossing point distribution. The first one is a crossing point distribution method. The crossing points on a face plane are divided into 36(6x6) sub-areas. The numbers of the crossing points in 36 sub-areas are used for the facial expression recognition. The second one is a displacement vector method. The size of an expression face is normalized to the size of a neutral face. Point on the facial surface with 3D-coordinate system is then converted to 2D-coordinate system. Two points on two faces which have the same 2D-coordinate are used to define a displacement vector. A displacement vector is a vector which points from a crossing point of a neutral face to a crossing point of an expression face. The two crossing points correspond to the pair point on the two faces. A surface on the expression face is divided to into 36(6x6) sub-areas based on the size of the neutral face. Displacement vectors in a sub-area are represented by an average of the displacement vectors. The 36 average displacement vectors are used for facial expression recognition.

In the chapter 4, experiments were done. The head pose variation in yaw and two situations of the head – with and without a cap were used. There were three experiments done. First, the nose ridge point was detected from a 2.5D partial face data set taken from any viewpoint ranging from -80 degrees to +80 degrees. The 2.5D partial face data were captured from 27 persons. All of the nose ridge from 223 samples with the cap can be

detected successfully. While ten samples fail in detection. Failure occurs because of the influence of messy hair. Second, the 3D face was reconstructed from a 2.5D partial face data set taken from any viewpoint ranging from -45 degrees to +45 degrees. The 2.5D partial face data were captured from 22 persons with and without a cap. A criterion defined by using a facial recognition was used to evaluate the reconstructed 3D face. By using the criterion, four samples of 187 samples were evaluated as an incorrect reconstruction. Third, the facial expressions were recognized by using a support vector machines technique and the radial basis function. 2.5D partial face data sets from 22 persons were taken from any viewpoint ranging from -45 degrees to +45 degrees. Four facial expressions which were smiling, surprise, anger and neutral were used. The experiments of the facial expression recognition were done based on the assumption that the person is known. The average recognition accuracy using the crossing point distribution scheme and the displacement vector scheme were 62.6% and 78.9% respectively. The advantages of our proposed method is that only one 2.5D partial face data set captured from any viewpoint of the face rotation between -45 degrees and + 45 degrees can be used. The results show the feasibility of the proposed method. In the chapter 5, the results of the proposed method were concluded.

Chapter 1 Introduction

As facial expressions provide important information such as affective stage, cognitive activity, temperament and personality, truthfulness, and psychopathology, the several researches study on facial perception over the last three decades[1),2)], but these research perform effectively for recognizing the facial expression from only full frontal facial image. However, in the case of many applications concerning robots, the partial face data from only one viewpoint is available. Therefore, it is important to develop a new innovation technology to meet the requirements of robots that use partial face data from only one viewpoint. This paper proposes a new method for facial expression reconstruction from a 2.5-dimensional (2.5D) partial face data set. A 2.5D data set consists of 2.5D range data and color data. The 2.5D data is a simplified three-dimensional (x, y, z) surface representation. This data contains at most one depth value (z-direction) for every point in the (x, y) plane. Each point in the (x, y) plane is associated with a registered texture color. Each scan can only provide a single viewpoint of the object instead of a full three-dimensional (3D) view[3)]. Therefore, some of the data of the facial surface is lost because it is not captured by the camera and not contained in the 2.5D face data set. However, the usage of a 2.5D image has advantages over a two-dimensional (2D) image when the face pose is varied. In our method, facial expressions were recognized by a 3D reconstructed face. A 2.5D partial face data set taken from any viewpoint between -45 degrees and +45 degrees was used to reconstruct a 3D face. Three important algorithms used in our method are nose ridge detection, 3D face reconstruction and facial expression recognition. The following methods have been proposed.

Nose ridge detection techniques commonly used in current research can be classified into three methods. Firstly, a shading technique was applied for identifying the location of the nose tip and the nose ridge[4),5)]. This method is limited because detecting the nose is sensitive to lighting conditions. Secondly, C. Hesher proposed a method based on the assumption that the nose tip was the highest point represented in the 3D facial data[6)]. Although this assumption has an advantage in computational time, it is not always satisfactory when used from any face orientation. Thirdly, a curvature technique can be adopted to define the nose location using 3D facial data[7)]. In our experiments involving

nose ridge detection, only half of the face can be captured from a side viewpoint. Therefore, the curvature technique was not appropriate for our experiment. Another method was presented by C. Xu, et al[8]. The nose tip is defined by means of effective energy (EE) based on the assumption that the nose tip is the highest local point and has a special shape like a peaked cup. This method has the sufficient accuracy result in case the complete shape of nose tip can be obtained. Actually, some part of nose may not be able to be obtained in some viewpoint. These existing nose ridge detection techniques provide sufficient results from only narrow angles of face orientation.

3D face reconstruction used in the current research can be classified into three techniques. Firstly, the 3D face is reconstructed from the multi images[9),10),11)]. The 2.5D image and the 2D image are applied in the existing research. G. Pujitha[12)] proposed a method using the four 2.5D images captured from different orientations. The corresponding points are selected manually and mapped by using a least squares estimator. The similar method which uses the four 2D images is also proposed in Ref. 13. The feature points on the front view image are mapped to the other three images. Based on the head pose and the mapped coordinate, the final depth value is calculated. L. Yuan and L. G. Marina[14)] proposed a method for reconstructing 3D face model from extracting 18 feature points on 2D facial images in two orthogonal views: one frontal view and one side view. Another type of face modelling approach is to create face models from video image[15),16)]. These methods need more than one single frontal image and the precision of depth value depend on the number of input images. X. Lu proposed a 3D face reconstruction method by integrating several 2.5D facial scans which were captured from different viewpoints. The feature points in Lu's method are extracted by matching two different viewpoints of the face based on a modified Iterative Closest Point (ICP) algorithm[3),17)]. K. Yamauchi used four compact range finders to obtain four view range images of the body for modelling all of the body including the face[18)]. Secondly, the morphable model proposed by V. Blanz[19),20),21)] can create a face model from a single image. This system uses both a geometry database and an image database. It can model the faces whose skin colours are covered by their database. Z. Liu.[22)] and D. Jiang[23)] proposed similar methods as [19,20]. Their methods are computationally expensive. To obtain the training set, a large 3D face

database is needed. Thirdly, shape from shading (SFS)[24),25),26)] deals with the recovery of 3D shape from a single monocular image. Under the Lambertian assumption, the 3D shape of human face can be recovered by using SFS. But most of the 3D face models recovered by SFS are not usable in many applications because they give unreliable depths in facial organs such as the eyebrows, eyes, mouth and nose. Another method is proposed by H. Yuankui[27)]. This method applied the generic face model with SFS. However, only the frontal image can be used to reconstruct the 3D face. Other techniques have also been proposed for reconstructing a 3D object[28),29),30)]. In these techniques, the ICP algorithm was used to match the surfaces of two 2.5D images. In these existing techniques, manual selection of feature points is needed. Furthermore, more than one 2.5D image is necessary for reconstructing a 3D face.

The following facial expression recognition methods have been proposed. Most of current researches are achievable for only frontal facial image. M. Pantic and L.J.M Rothkantz[31)] proposed a facial expression recognition method by defining their face model as a point-based model composed of the two 2D facial views of the frontal and side view face images. Multiple feature detectors were applied redundantly in order to localize contours of prominent facial features prior to their model. This method needs more than one viewpoint of face image including frontal face image. H., Michael et al.[32)] use a generic 3D head model and frontal view image for the synthesis of facial expression, and then the facial expression was classified by local binary pattern techniques. I.A. Essa[33)] adopts the optical flow processing for perception and measurement of facial motion. The 3D mesh model of face is fitted by 2D frontal face image. Muscle activation is extracted using a physics-based model of facial deformation, and then the deformation is estimated from the optical flow. C. Li and A. Barreto[34)] proposed the framework of 3D face recognition involving an initial expression assessment of unknown face. The smiling and neutral expressions were recognized. The most distinctive features associated with the smile are the bulging of the cheek muscle and the uplift of the corner of the mouth, and then the histograms of the range value (z-coordinates) of cheeks from smiling and neutral face is used for recognition by means of Linear Discriminant Analysis (LDA) and Support Vector Machine (SVM). J. Wang et al.[35)] used the rotated 3D frontal face model under

viewpoint between -/+40° for each rotation on pitch and yaw rotation to recognize facial expression. Based on the principal curvature information estimated on the 3D triangle mesh model, a surface labeling approach is applied to classify the 3D primitive surface features into twelve basic categories. The statistic histograms of the surface labels of all these regions are combined to construct the specific facial expression feature. Linear Discriminant Analysis (LDA) is used for classification. In this method, the rotated frontal face is used in the experiment but, actually, an only partial face image can be obtained in case of capturing the face from side view. In the current research of facial expression recognition, the frontal face image is necessary to achieve the satisfactory results. However, the orientation of face is also a factor to make the recognition fail.

This paper proposes a novel method for facial expression recognition from a 2.5D partial face data set. The 2.5D partial face data set is captured from any viewpoint ranging from -45 degrees to +45 degrees around y-axis (see Fig. 1). Our method is developed for recognizing four facial expressions which are neutral, anger, surprise and smiling. The proposed method includes three new technical components for nose ridge detection, 3D face reconstruction and facial expression recognition.

The first new technical component consists of an ellipse fitting technique that fits the partial data of a cross section of the head. Using this fitting technique, the center of the head and the semi-major axis can be obtained. Consequently, the nose ridge can be roughly detected as the farthest point from the center of the head. Secondly, the detected nose ridge and the center of the head are corrected iteratively to obtain a higher precision. From this refining procedure, the symmetry plane of the head can be obtained. The 3D virtual expression face is then reconstructed using this symmetry plane (see Fig. 1(b)). Thirdly, the facial skin of the virtual face is used for facial expression recognition by using a face plane algorithm (see Fig. 1(c,d)). Facial expression is represented in terms of the crossing point distribution and the displacement vector on the face plane (see Fig. 1(e,f)). By using a support vector machines method, the facial expressions are then recognized.

In our method, the nose ridge can be detected from only one 2.5D partial face data set. This data set can be captured from any viewpoint ranging from -80 degrees to +80 degrees around the y-axis. In the 3D face reconstruction and facial expression recognition, only one

2.5D partial face data set is captured from any viewpoint ranging from -45 degrees to +45 degrees because over half the area of the face is used to evaluate the symmetry plane.

The advantages of our nose ridge detection algorithm are the following: 1) It is not sensitive to lighting conditions, 2) It can overcome noisy data, 3) It is reliable enough to localize the nose position even when only half of the face can be captured.

In the following chapters, the algorithms for the nose ridge detection and for the 3D face reconstruction are described in chapter 2. In chapter 3, the facial expression recognition algorithm is described. The experimental results are shown and discussed in chapter 4 and 5 respectively.

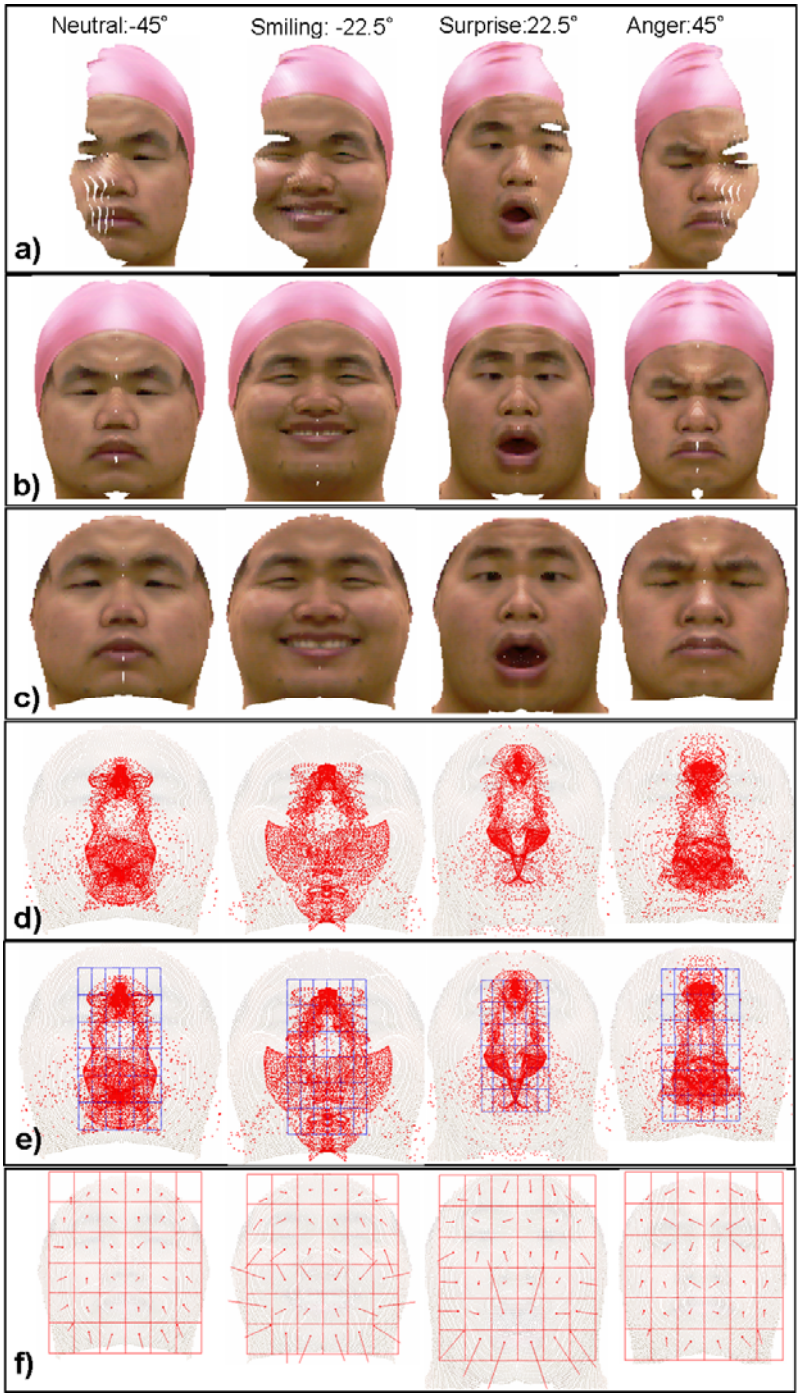

Figure 1. a) The input 2.5D partial face images are captured in different viewpoint starting from -45° (Neutral), -22.5° (Smiling), 22.5° (Surprise) and 45° (Anger). b) The 3D virtual expression face is reconstructed. c) Only face surface is extracted to be used in face plane algorithm. d) Crossing point on face plane. e) The crossing point distributions on the face plane and the face profile. f) A displacement vector set of each expression face consisting of 36 average vectors are used to analyze the facial expression.

Chapter 2 3D Face Reconstruction

For the facial expression recognition method, a 3D reconstructed face is used. In this chapter, the 3D virtual face is reconstructed from a 2.5D partial face data set. The 3D face reconstruction algorithm is described.

To reconstruct a 3D face, the 2.5D partial face data set is sprit into N cross sections along Y axis. The data on each cross section was fitted by ellipse fitting technique. The nose tip and the nose ridge are then detected by correcting the face vectors according to nose ridge detection algorithm. The symmetry plane passing the nose tip, the nose ridge and the point on the center line of the face is used for reconstructing 3D face according to the symmetry plane correction algorithm (see Fig. 1(b)).

A flowchart of our 3D face reconstruction algorithm is shown in Fig. 2. The 2.5D image acquisition, nose ridge detection and symmetry plane correction are explained in the following subsections.

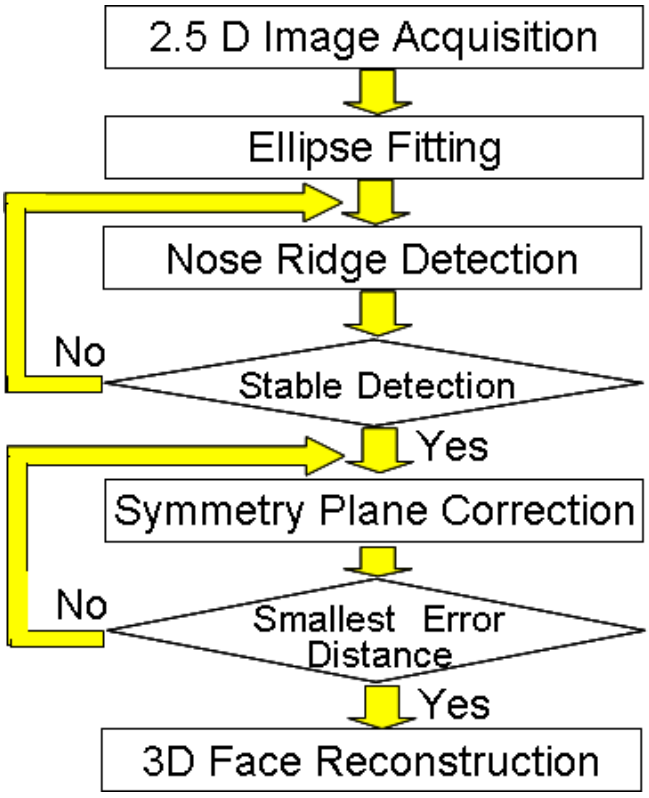

Figure. 2. Flowchart of our 3D face reconstruction algorithm.

2.1. 2.5D Image Acquisition

For our 3D face reconstruction algorithm, a single 2.5D head data set is used. The input data is captured by using a VIVID700 scanner and represented by point set data (see Fig. 3). The 2.5D data consists of range data and color data (see Fig. 4). The range image size is 200x200 pixels, the color image has 400x400 pixels in the same scanning region. The object stands in front of the scanner at a distance of about 1.5 m as shown in Fig. 5. Unfortunately, the VIVID700 scanner is not sensitive to black colors such as the kind found typically in Asian hair because it uses a red laser. This problem can be solved by covering the hair with a cap. It seems to be advantageous to wear a cap. Therefore, the head without the cap is also tested by changing the hair color. The different results in each case will be discussed in our experiment. The input image, in both cases, was captured from many viewpoints around the y-axis varying from -80 degrees to +80 degrees to detect the location of the nose tip and the nose ridge. (see Fig. 6)

Figure 3. 3D digital scanner: Vivid 700

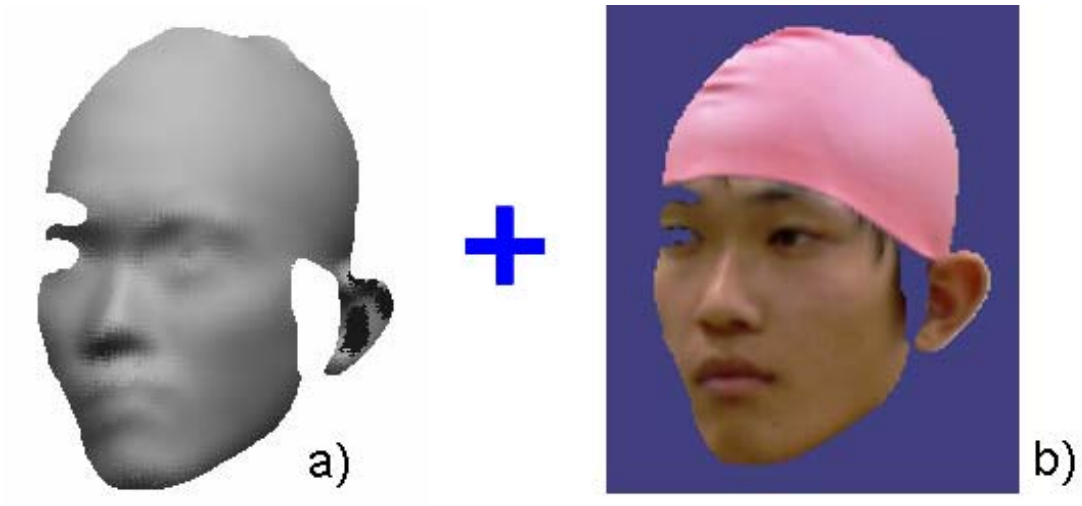

Figure 4. The 2.5D data consists of a) range image and b) color data

Figure 5. The object stands in front of the scanner at a distance of about 1.5 m. The camera on the right hand in the figure is VIVID700

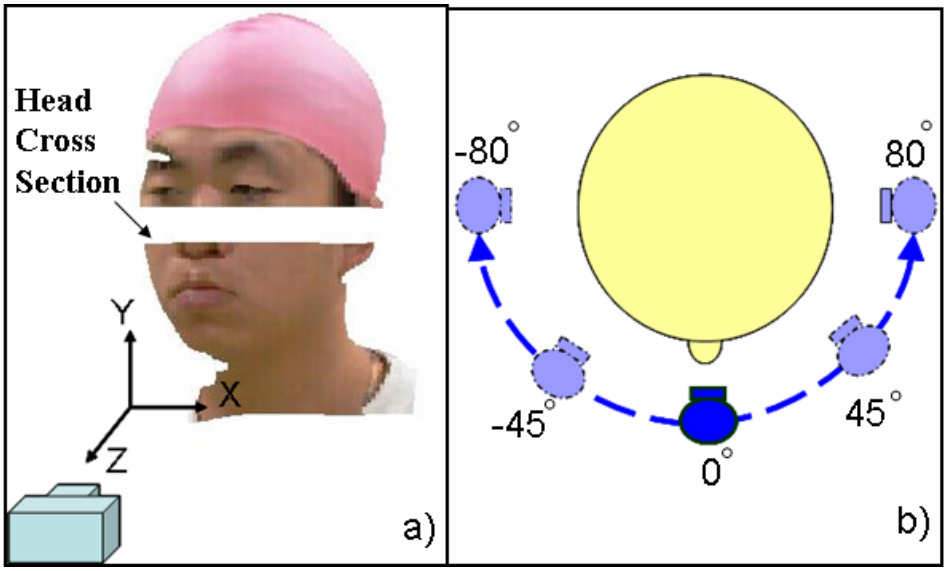

Figure 6. a) Coordinate System Configuration. The black color surface as hair unmeasured by laser light is covered by cap. b) Illustrate the viewpoint for capturing the 2.5D image from the top view of human head.

```
1   Function a = fit_ellipse(x,y)
2   D1 = [x .^ 2, x .*y, y.^2];                 % quadratic part of the design matrix
3   D2 = [x, y, ones(size(x))];                 % linear part of the design matrix
4   S1 = D1'*D1;                                % quadratic part of the scatter matrix
5   S2 = D1'*D2;                                % combined part of the scatter matrix
6   S3 = D2'*D2;                                % linear part of the scatter matrix
7   T = -inv(S3)*S2';                           % for getting a2 from a1
8   M = S1 + S2*T;                              % reduced scatter matrix
9   M = [M(3,:) ./ 2; -M(2,:); M(1,:) ./ 2];    % Premuliply by inv(C1)
10  [evec, eval] = eig(M);                      % solve eigensystem
11  cond = 4 * evec(1,:) .*evec(3,:) -evec(2,:) .^ 2 ; % evaluate a'Ca
12  a1 = evec(:,find(cond > 0));                % eigenvector for min. pos. eigenvalue
13  a  = [a1; T*a1];                            % ellipse coefficients
```

Figure 7. MATLAB implementation of the ellipse-specific fitting algorithm in Ref. 36

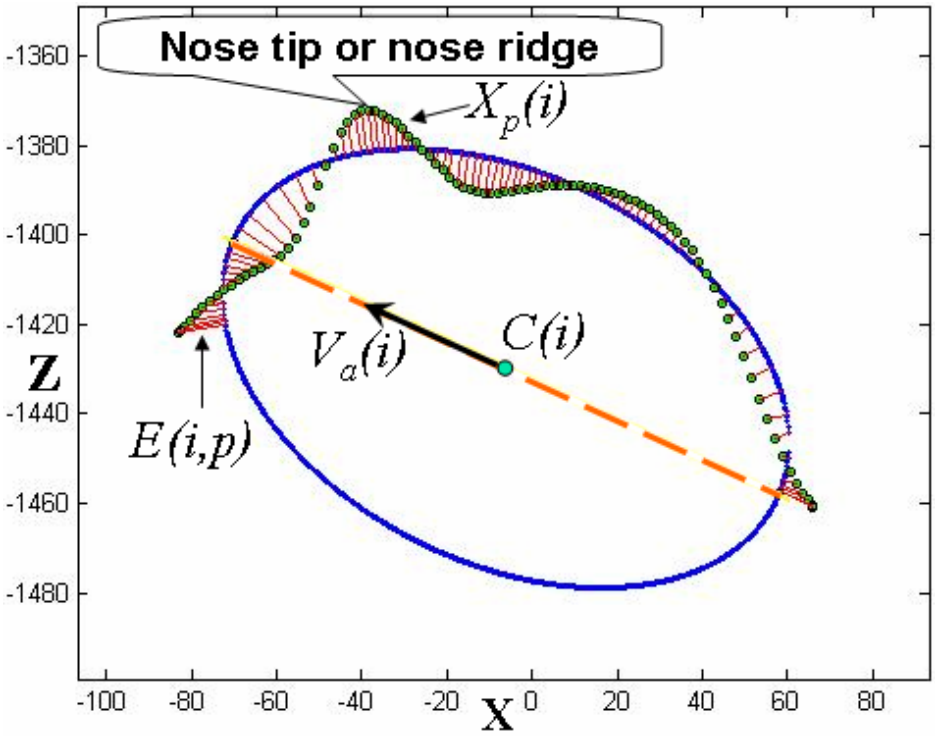

Figure. 8. The head cross section data passing the nose are fitted by an ellipse fitting technique. $X_p(i)$ is a point on the facial surface. The lengths of the lines from the points on the facial surface to the ellipse are the ellipse fitting errors $E(i,p)$ of the points $X_p(i)$.

2.2. Ellipse Fitting Technique

2.2.1. Fitting of Ellipse for Head Cross Section Data

In this subsection, three important parameters which can be obtained by the ellipse fitting technique are described. The head data from the range image is split into N cross sections parallel with the xz-plane. The data on each cross section are fitted by using an ellipse fitting technique[36]. The conic equation (Eq. (1)) of the ellipse can be described in the xz-plane according to the camera coordinate system:

$$F_i(x,z)=ax^2+bxz+cz^2+dx+ez+f=0 \quad (1)$$

with an ellipse-specific constraint:

$$b^2-4ac<0 \quad (2)$$

where $F_i(x,z)$ is the ellipse equation of the i^{th} cross section. The parameters a, b, c, d, e and f are coefficients of the ellipse, and (x,z) is the set of coordinates lying on the i^{th} cross

section: $i=1,\ldots,N$. The ellipse-specific fitting algorithm proposed in Ref. 36 is shown in fig. 4.

The three important parameters illustrated in Fig. 8 can be obtained as follows:

i) The Average Center of the Ellipse (C_{Ae}): This parameter is the average value computed from the centers of the ellipses $C(i)$ in all cross sections as shown in Fig. 9 and Eq. 3).

$$C_{Ae} = \frac{1}{N}\sum_{i=1}^{N} C(i) \tag{3}$$

where $C(i)$ is the center of ellipse of i^{th} cross section. This average center point is decided tentatively. It will be corrected in subsection 2.5.

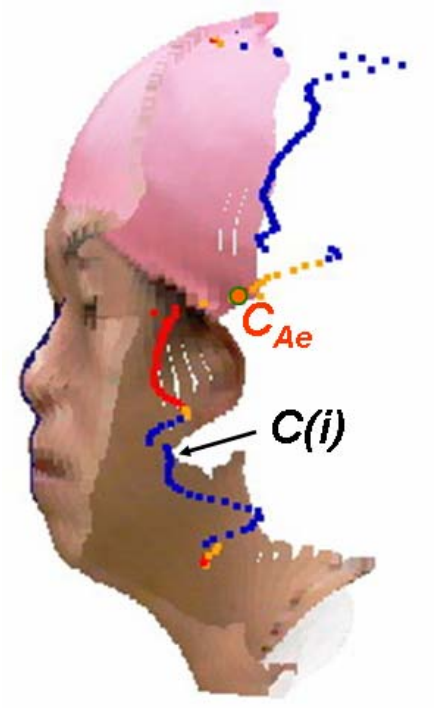

Figure 9. The centers of the ellipses $C(i)$ in each cross section of 2.5D partial face data in the side viewpoint. C_{Ae} is the average of centers of the ellipse.

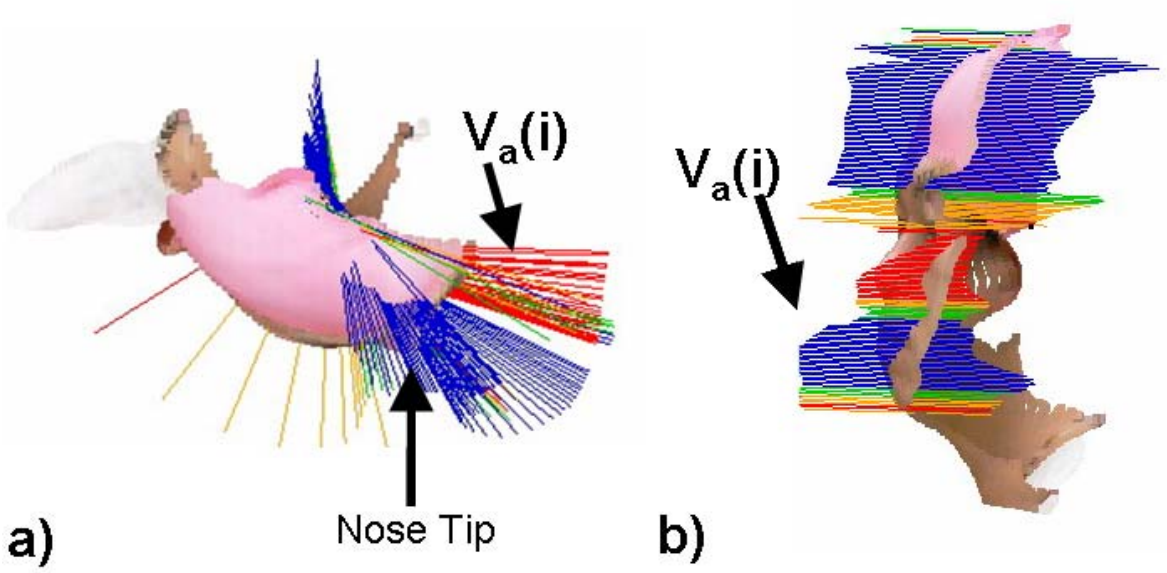

Figure 10. a) The semi-major axis vectors from the top view of a partial face. b) The semi-major axis vectors from the side view of a partial face.

ii) The Semi-major Axis Vectors $V_a(i)$ of the i^{th} Cross Section: There are two opposite directions for the vector along the semi-major axis of the ellipse in each cross section pointing away from the center. In each cross section, the semi-major axis vector pointing to the facial surface is selected as shown in Fig. 8. The direction of the semi-major axis vectors of all cross sections in head region are shown in Fig. 10.

iii) The Average Ellipse Fitting Error $E_{Ae}(i)$: The ellipse fitting error $E(i,p)$ at point $X_p(i)= (x_p,z_p)$ in the i^{th} cross section is defined as shown in Fig. 8, where p is a point in the cross section data. $E(i,p)$ is the orthogonal distance between point $X_p(i)$ and the ellipse. $E_{Ae}(i)$ is the average of all $E(i,p)$ for the i^{th} cross section. The $E_{Ae}(i)$ is computed from $E(i,p)$ as shown in Eq. (4).

$$E_{Ae}(i) = \frac{1}{M}\sum_{p=1}^{M} E(i,p) \tag{4}$$

where M is the total number of the points on the i^{th} cross section.

2.2.2. Cross Sections in the Nose Region

The nose ridge is detected from the cross sections in the nose region. According to a

pilot test we ran, there are fifteen cross sections in the nose region counting from the cross section passing to the nose tip as shown in Fig. 11. Therefore, the nose region can be determined by detecting the nose tip. As an empirical observation, the cross section including the nose tip always has the maximum average ellipse fitting error.

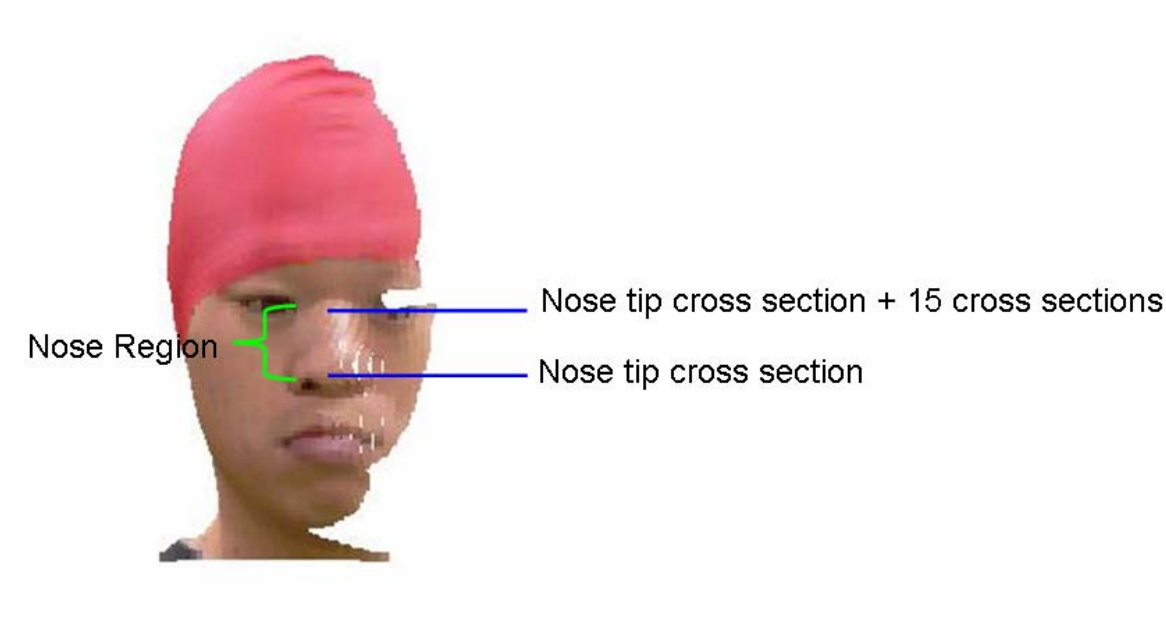

Figure 11. Nose region consist of 15 cross sections start counting from the cross section passing the nose tip.

2.3. Nose Ridge Detection

In subsection 2.2., the head cross section was fitted by ellipses and the three significant parameters have been computed. This subsection describes the nose ridge detection algorithm.

2.3.1. Face Vector Calculation

As mentioned in 2.2.1(ii), there are many semi-major axis vectors pointing to the facial surface. Based on the experiments of all viewpoints, each cross section shows the different average ellipse fitting error. According to this property, the semi-major axis vector under the condition that the average ellipse fitting error of its cross section is less than 1 mm. tend to point to the surface near the nose. By averaging the semi-major axis vectors obtained from one viewpoint, a face vector can be calculated, and then the projected face vector (V_{pf}) which is the projection of the face vector onto each cross section in nose region

will be corrected to the nose tip or ridge in the subsection 2.3.2.

In the case of front viewpoint, the direction of semi-major axis vectors under the condition is unstable. In this case, the direction of the projected face vector cannot be corrected to the nose tip or ridge properly. Therefore, the following algorithm is proposed to cope with this problem.

In the nose region, the facial surface was divided into three zones equally by 60 degrees. The average of the ellipse fitting error was computed zone by zone. The three average errors of each zone were compared with each other. The average ellipse fitting error in the middle zone shows the biggest error in case the nose or ear exists in the middle zone (see Fig. 12).

The existence of nose can be evaluated by counting the number of semi-major axis vectors which point to around the center of the middle zone. If the number of the semi-major axis vectors is more than a criterion, it means that the nose exists in the middle zone. The results are satisfactory when the criterion is 12. In case of the nose located in the middle zone, the face vector is computed from the average of semi-major axis vectors which point to around the center of the middle zone, and then the projected face vector can be corrected to the nose tip or ridge according to the algorithm in the subsection 2.3.2.

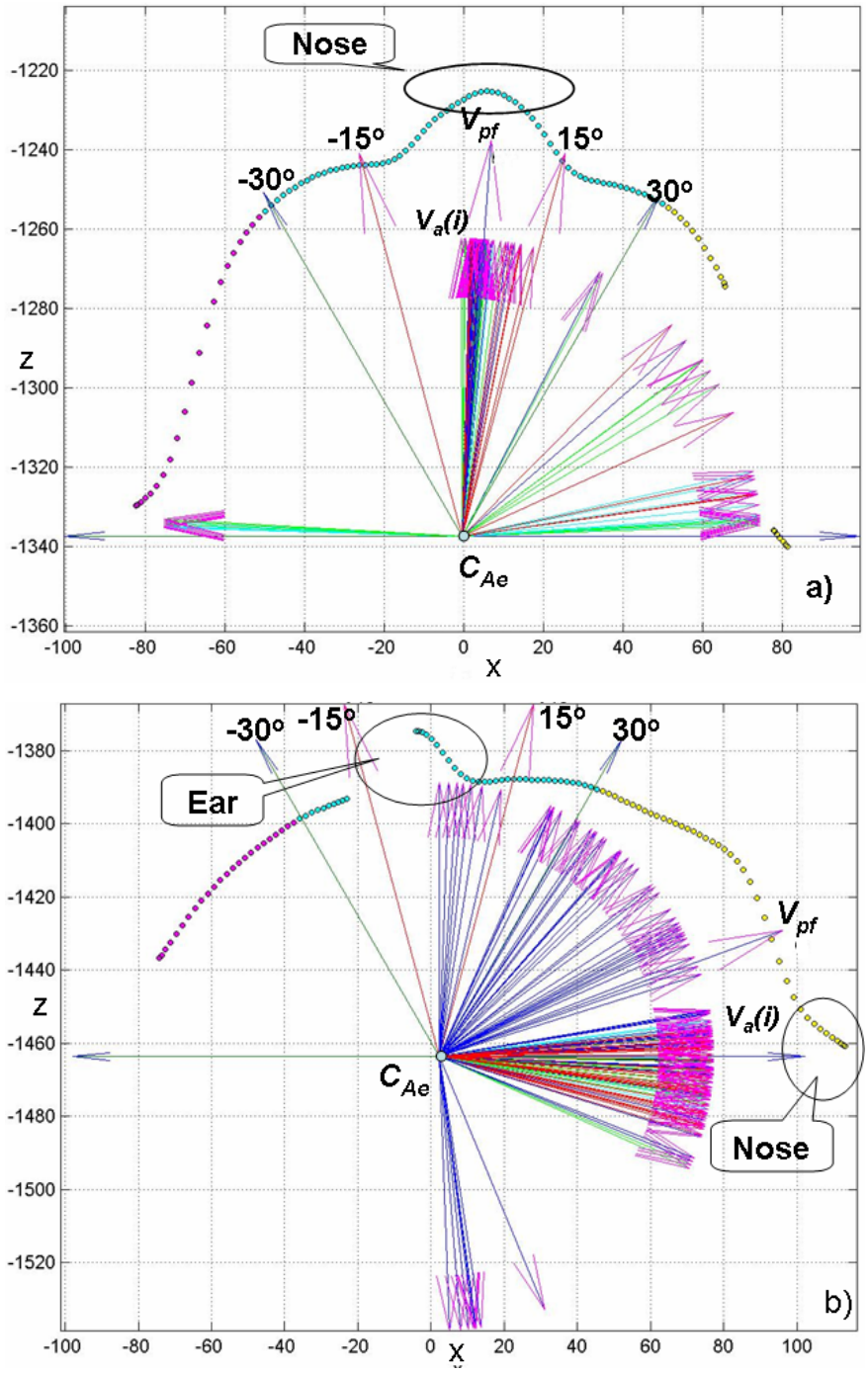

Figure 12. The nose and ear show the biggest average error in the middle zone in (a) and (b), respectively. V_{pf} is the projected face vector. $V_a(i)$ is the semi-major axis vectors. a) The head cross section of viewpoint 0 degrees is illustrated. The most of vectors $V_a(i)$ point near the nose. b) The head cross section of viewpoint -80 degrees. The many of vectors $V_a(i)$ does not point to the nose.

2.3.2. Face Vector Correction Algorithm

The face vector is obtained by averaging the semi-major axis vector according to the algorithm in 2.3.1. However, the face vector does not always pass through the nose tip. Therefore, a detection algorithm is needed. The following pseudo code explains the nose ridge detection algorithm.

[Nose ridge detection algorithm]

project the face vector onto each cross section of the nose region; // (projected face vector V_{pf})

set an initial value for the detected point;

for each cross section of the nose region {

do {

project the point set on facial surface (P_s) onto the projected face vector; // (see Fig. 13(a))

find the projected point which shows the farthest distance from the average center point C_{Ae};

correct the projected face vector to the point which shows the farthest distance;

} while (the detected point is not the same with the previous point)

}

[End of algorithm]

This process is applied to all of the 15 cross sections in the nose region. The 15 nose ridge points are determined tentatively because the average center point C_{Ae} is also determined temporarily. The nose ridge vector (V_{nr}) can be obtained as the eigenvector with the highest eigenvalue computed from the 15 nose ridge points by using a principal components analysis technique. The vector pointing from the average center point C_{Ae} to the center of the 15 nose ridge points is then defined as nose vector (V_{no}) (Fig. 13(b)). In subsection 2.4, the mirror face construction method will be explained. The correction of V_{no} and C_{Ae} will be discussed in subsection 2.5.

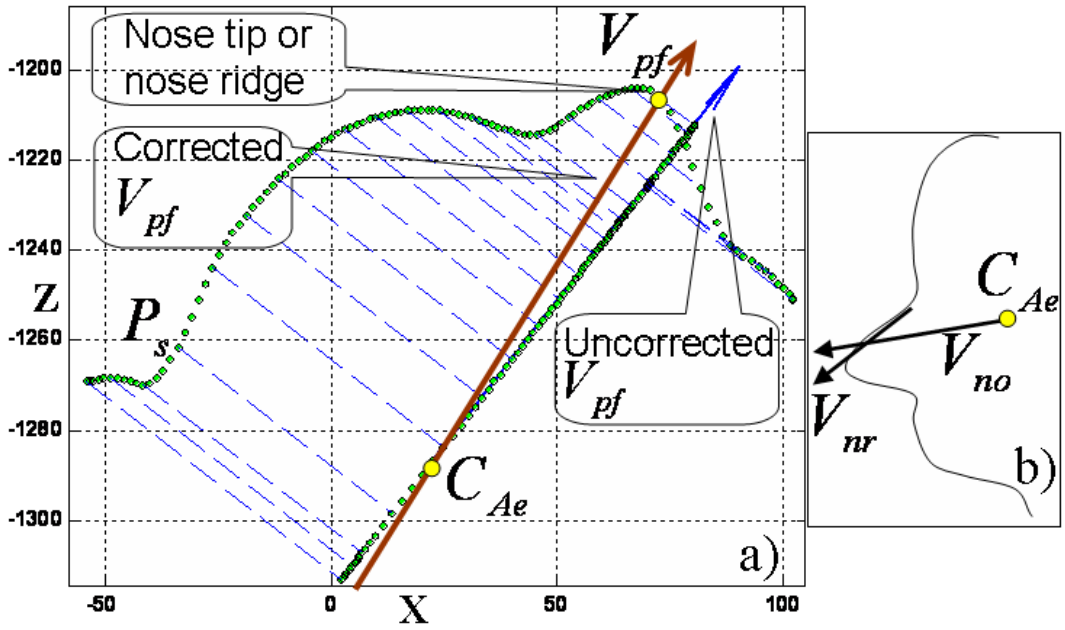

Figure. 13. a) The projected face vector V_{pf} points from the average center point C_{Ae} to the nose ridge point. P_s is a points set on the facial surface. The dashed line illustrates the projection of P_s onto V_{pf}. b) The two vectors in a face side view are the nose ridge vector V_{nr} pointing along the nose ridge points, and the nose vector V_{no} pointing away from the average center of the ellipse C_{Ae}.

2.4. Mirror Face Construction

The "mirror face" is calculated from the real face, and then the virtual face is reconstructed from the real face and its mirror face. The mirror face is computed by reflecting the real face in the symmetry plane. In this subsection, the symmetry plane estimation and the mirror face computation is described.

Due to the bilateral property of the face[37], the symmetry plane, which is called "midsagittal plane" in face anatomy, is defined as the vertical plane along the center line of the face through the nose tip and the nose ridge. The symmetry plane is defined by using two vectors: the nose ridge vector and the nose vector.

In subsection 2.3.2., the 15 nose ridge points were detected. The nose ridge vector can be calculated from these points by using a Principle Components Analysis method (PCA) (see Fig. 14(a)).

The nose vector *(V_{no})* is the vector pointing from the average center C_{Ae} to the center of 15 nose ridge points (see Fig. 14(b)).

According to the nose ridge vector and the nose vector, the symmetry plane is described in Eq. 5 and 6.

$$V_n = V_{nr} \times V_{no} \tag{5}$$

$$V_n \bullet X = 0 \tag{6}$$

where V_{nr} is the nose ridge vector, V_{no} is the nose vector, V_n is the normal vector of symmetry plane and X is a point on the symmetry plane, where the origin is C_{Ae} and $\times$ and $\bullet$ represent outer product and inner product respectively.

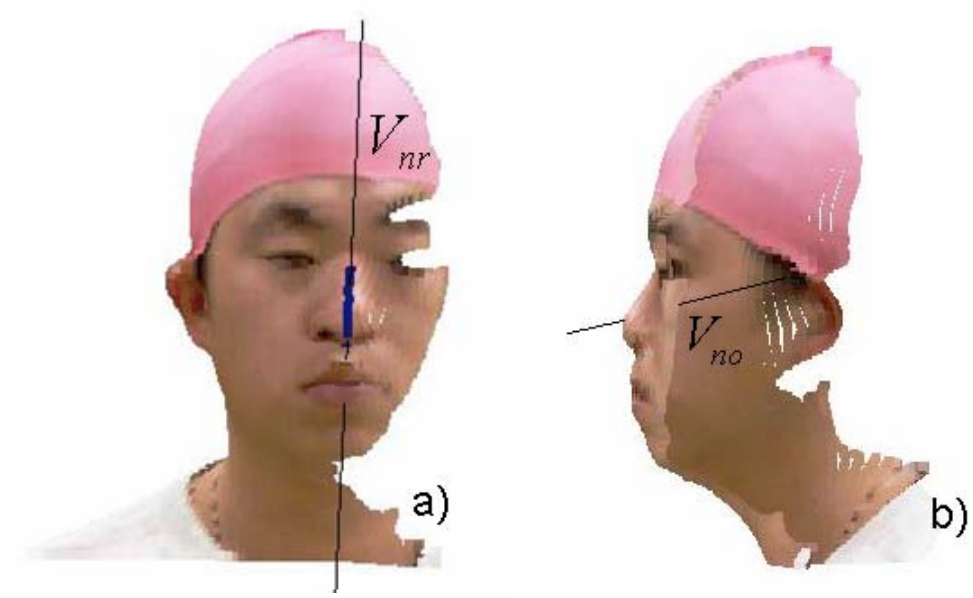

Figure. 14. a.) The nose ridge vector *(V_{nr})* passes the nose ridge points. b.) Illustrate the side viewpoint of real face, the nose vector *(V_{no})* point from the average center of head to the average center of nose ridge points.

In previous paragraph, the symmetry plane can be estimated. By reflecting the real face in the symmetry plane, the mirror face can be computed.

The vector (V_r) pointing from the average center (C_{Ae}) to a point on the real face is divided into two components which are perpendicular one (V_e) and parallel one (V_a) for the symmetry plane. The vector (V_f) pointing from the average center (C_{Ae}) to the point on the mirror face is combined with the inverse of perpendicular vector ($-V_e$) and parallel

vector (See Eq. 9, Fig. 15).

$$V_e = (V_r \bullet V_n) * \frac{V_n}{|V_n|^2} \tag{7}$$

$$V_a = V_r - V_e \tag{8}$$

$$V_f = -V_e + V_a \tag{9}$$

where V_n is the normal vector as described in Eq. 5.

As mentioned above, the mirror face can be computed.

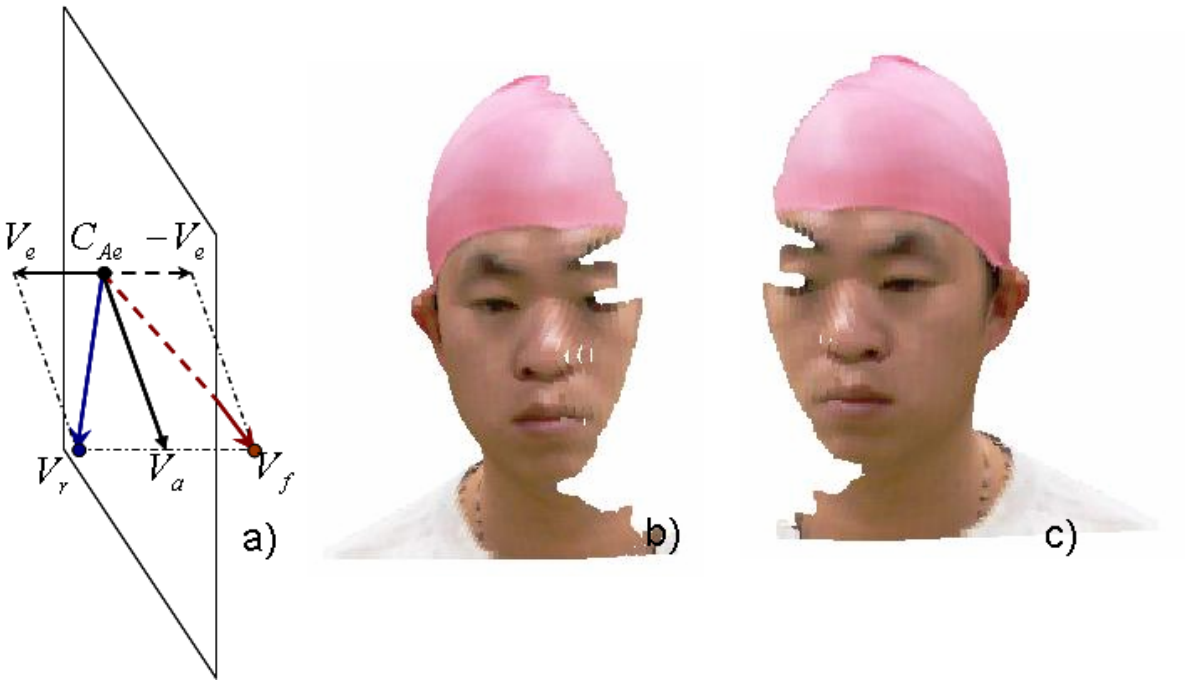

Figure. 15. a) Component vectors in Eq. (7)-(9) with the symmetry plane. The symmetry plane is computed from the nose ridge vector and nose vector. V_r points to a point on the real face. V_e is the perpendicular vector. V_a is the parallel vector to the symmetry plane. V_f points to the mirror point on the mirror face. b) The front side of real face from 22.5 degrees c) The front side of the mirror face from 22.5 degrees.

2.5. Symmetry Plane Correction

The virtual face is reconstructed according to the algorithm in 2.4. However, some virtual faces result in faces that are either too fat or too thin. These problems occur due to the incorrect location of the center point of the head. In this subsection, a correction algorithm that determines the average center point C_{Ae} lying on the nose vector and the extended nose ridge vector will be explained.

2.5.1. Correction of the Average Center C_{Ae}

This subsection, the correction of the average center C_{Ae} will be explained. The nose vector is rotated around the nose ridge (Fig. 16.) by an average angle computed from the angle between the two vectors V_{am} and V_{ar}. V_{am} and V_{ar} are the vectors pointing from the nose ridge point to the corresponding points on the mirror face and the real face respectively. The corresponding points are the closest points on the real face and the mirror face. The nose vector V_{no} is rotated by the rotation angle $\theta/2$ centered at the nose ridge point. The angle θ is the average of α_i which is the angle between V_{ami} and V_{ari}. The average center point C_{Ae} is moved to a new position C'_{Ae} by rotating the nose vector. The symmetry plane and the mirror face are then computed again. The average center point is more stable when the average error distance between the corresponding points on the mirror face and the real face is smaller than the previous iteration. After the center point C_{Ae} has moved to the new center point C'_{Ae}, the farthest point from the center point C_{Ae} is not farthest from the new center point C'_{Ae}. Therefore, the nose ridge which is the farthest point from C'_{Ae}, is detected again by using the nose ridge detection algorithm described in subsection 2.3. This algorithm is repeated until the average error distance shows a bigger value than the previous one.

This algorithm is illustrated by the following pseudo code:

[Symmetry plane correction algorithm]

calculate the average distance error between the mirror face and the real face;

set an initial value for the average distance error;

do {

calculate θ *using two vectors* V_{am} *and* V_{ar}*;*

move C_{Ae} *to* $C_{Ae}^{'}$ *by rotating the nose vector with angle* $\theta/2$*;*

make the symmetry plane and calculate the mirror face;

calculate the average distance error between the mirror face and the real face;

apply the nose ridge detection algorithm described in subsection 2.3;

} while (the average distance error is smaller than the previous one)

[End of algorithm]

Thus, the average center point C_{Ae} and the nose ridge can be corrected at the same time. The convergence of this algorithm was confirmed by our experiment (Fig.17). By using the symmetry plane correction algorithm, the 3D virtual face is reconstructed.

2.5.2. Extended Nose Ridge Vector Computation

According to the inclination problem of the nose ridge vector, the extended nose ridge vector is proposed for solving the inclination problem of the symmetry plane which is the consequence of the nose ridge vector.

A) Region of Interest. The region of interest (ROI) for calculating the extended nose ridge vector is defined as the cross section between the average center cross section and the chin neck cross section. The chin neck cross section is the cross section passing the chin neck junction.

B) Farthest Point Detection. The points set on facial surface (P_s) in ROI are projected on the nose vector. Generally, the farthest point from the average center C_{Ae} is on the center line of face. The algorithm in 2.3.2 is used to detect the farthest point. The process is done for the whole cross sections in ROI. The farthest points are detected finally. The extended nose ridge vector is computed from the farthest points by means of the PCA. The symmetry plane can be computed from the extended nose ridge vector and the nose vector.

C) Outlier Elimination. Because there are some detected farthest points not lying on the center line of face, these points have an effect on the inclination of the extended nose ridge vector and the inclination of the symmetry plane. These points should be eliminated.

The standard deviation of all distances from the detected farthest points to the symmetry plane is calculated. The point which has the distance bigger than the standard deviation is hen removed (see Fig. 18(a)). Finally, the extended nose ridge vector is computed from the remaining farthest points by the PCA.

After the correction algorithm is done, the symmetry plane is computed from the extended nose ridge vector and the nose vector, and then the mirror face is also computed by reflecting the real face in the symmetry plane as mentioned in 2.4. Finally, the 3D face is reconstructed as shown in Fig. 18(b).

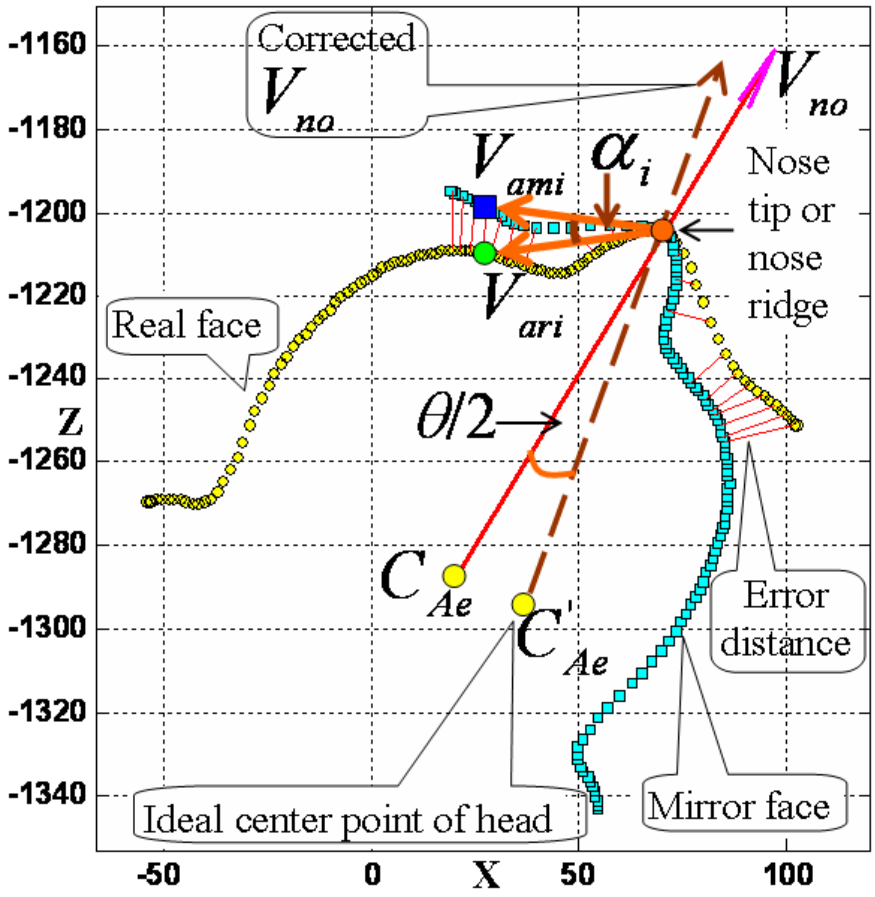

Figure. 16. C'_{Ae} is the new center instead of C_{Ae} after rotating V_{no}. The rotation angle $\theta/2$ corresponds to α_i for correcting the nose vector.

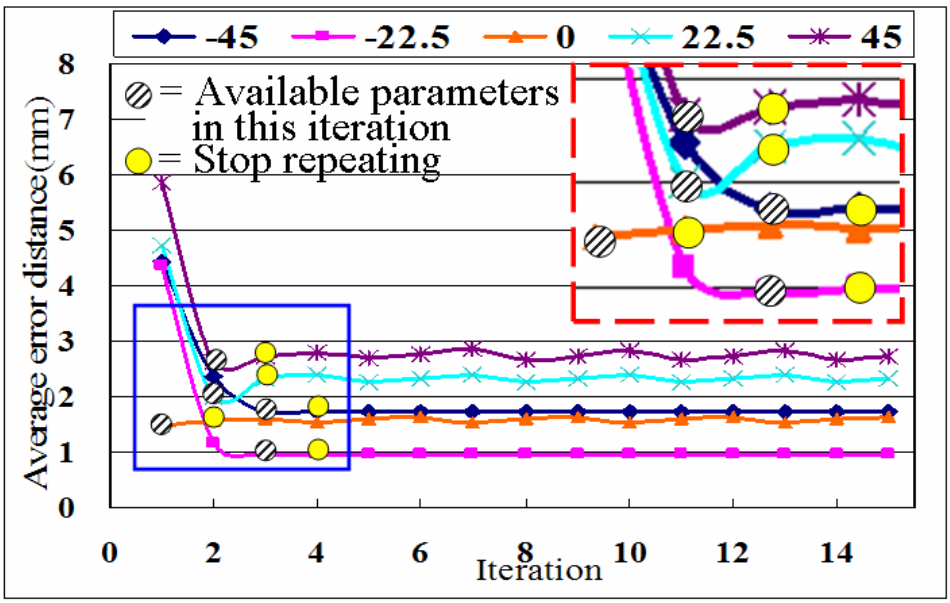

Figure. 17. The convergence of the average error distance over 15 iterations in the symmetry plane correction algorithm. If the average error distance increases compared to the previous iteration, the algorithm will stop and the parameter computed in the previous iteration will be used in the next part of algorithm. The graph in the red dash frame is an enlargement of the blue frame.

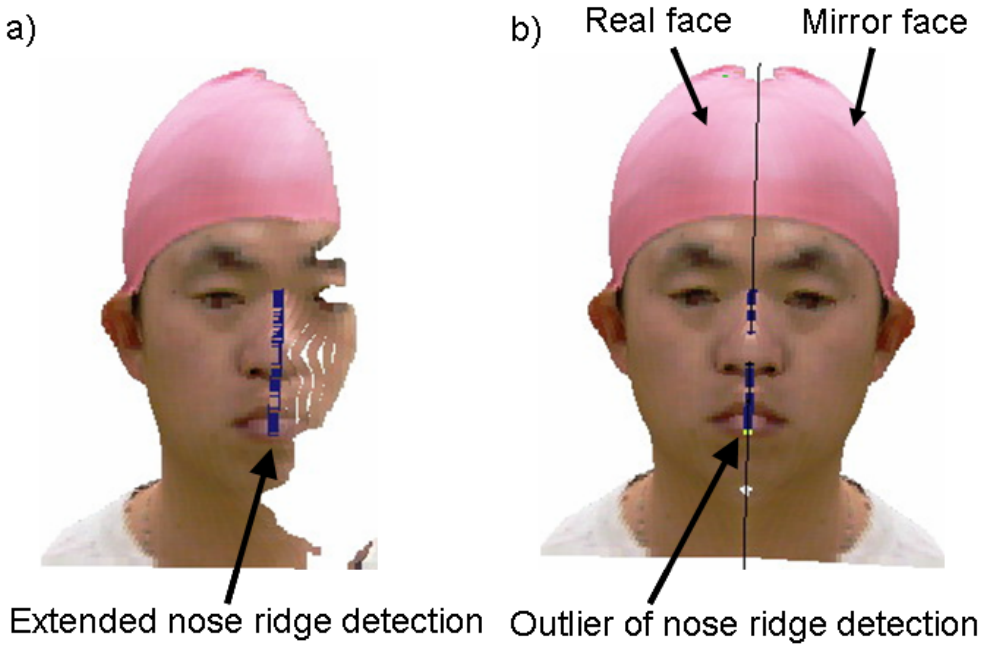

Figure. 18. a) The extended nose ridge detection. b) The reconstructed 3D face. The yellow point is the outlier of the nose ridge which is removed.

Chapter 3 Facial Expression Recognition

In the facial expression recognition algorithm, expression of the face is represented as displacement vectors by a face plane algorithm. Crossing points on the face plane are use to compute the displacement vectors. In this chapter, the algorithms for computing a face plane and two crossing point analysis methods, crossing point distribution and displacement vectors, are explained. A flowchart of facial expression recognition algorithm is shown in Fig. 19.

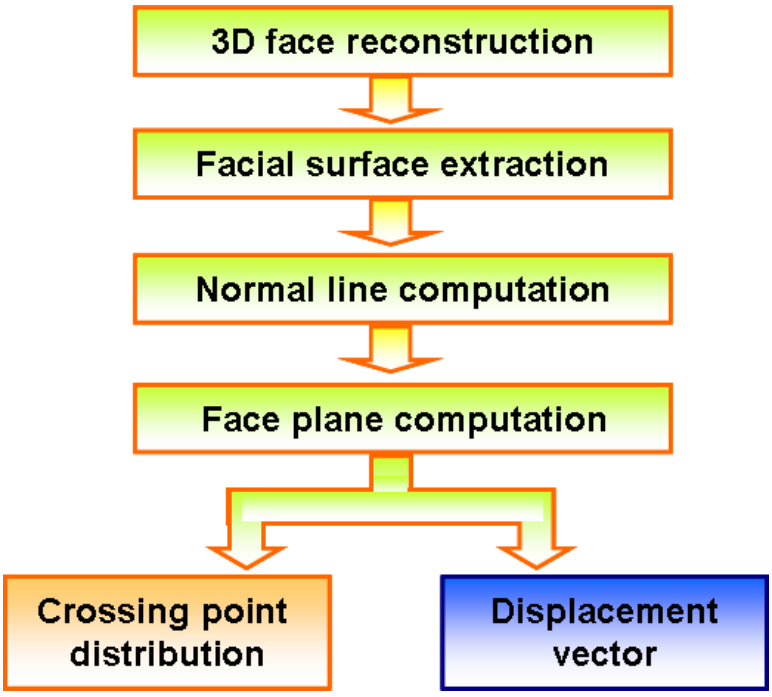

Figure 19. Flowchart of facial expression recognition algorithm.

3.1. Face Plane Computation

After a virtual face was reconstructed in chapter 2, facial surface data extracted by using color segmentation according to a Lab color space as shown in Fig. 1(c) are used in this subsection. A face plane algorithm is applied to represent facial expression in terms of crossing points[38]. The advantage of the face plane is that the relative position from the face is fixed.

3.1.1. Facial Surface Extraction

As only the facial surface data excluding the hair or cap data for is used to compute

the face plane, in this subsection, the color segmentation method to extract the data of the facial surface is explained. A flowchart of facial surface extraction algorithm is shown in Fig. 20.

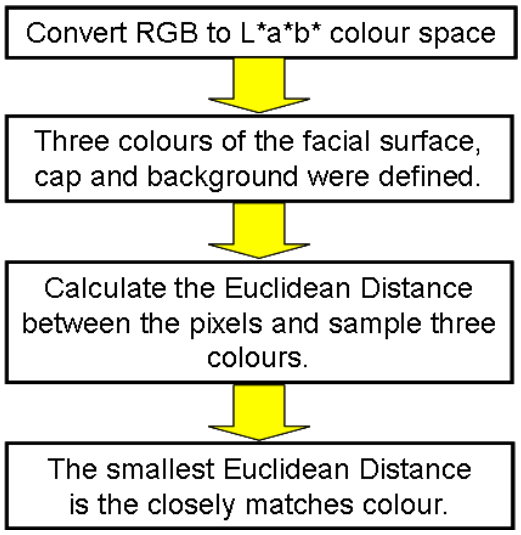

Figure 20. A flowchart of facial surface extraction algorithm

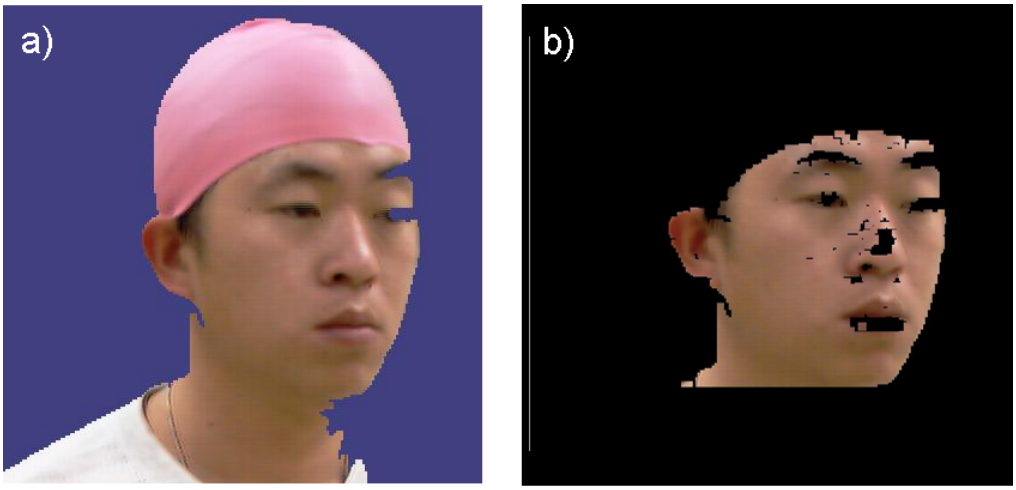

Figure 21. a) The color image before segmenting the facial surface. The three main objects are a facial surface, cap and background. b) The facial surface after segmentation.

The Euclidean distance is used to classify each pixel in a face color image in L*a*b* color space. In a face color image, there are three average a* and b* value from face surface, a cap and background as shown in Fig. 21. By using the average a* and b* values, each pixel in the image is classified by computing the Euclidean distance between that

pixel and each average color of each object. The pixel will be classified into a group of the object that show the smallest distance. The pixels in the group of the facial surface will be used to compute the face plane in next subsection.

The RGB color space is translated into the uniform color space. The CIE L*a*b* and L*u*v* color spaces are popular in the image processing field[39]. The quantities L*, a*, and b* are defined by Eq. 10:

$$L^* = \begin{cases} 116(Y/Y_0)^{1/3} - 16 & if \quad (Y/Y_0) > 0.008856 \\ 903.29(Y/Y_0) & else \end{cases}$$
$$a^* = 500[(X/X_0)^{1/3} - (Y/Y_0)^{1/3}] \tag{10}$$
$$b^* = 200[(Y/Y_0)^{1/3} - (Z/Z_0)^{1/3}]$$

where the constants X_0, Y_0 and Z_0 are the tristimulus values of the standard white. Also, the value X, Y, and Z are defined using R, G, and B values as follows:

$$X = 0.490R + 0.310G + 0.200B$$
$$Y = 0.177R + 0.812G + 0.011B \tag{11}$$
$$Z = 0.000R + 0.010G + 0.990B$$

The component L* represents the intensity. A combination of a* and b* represents a color.

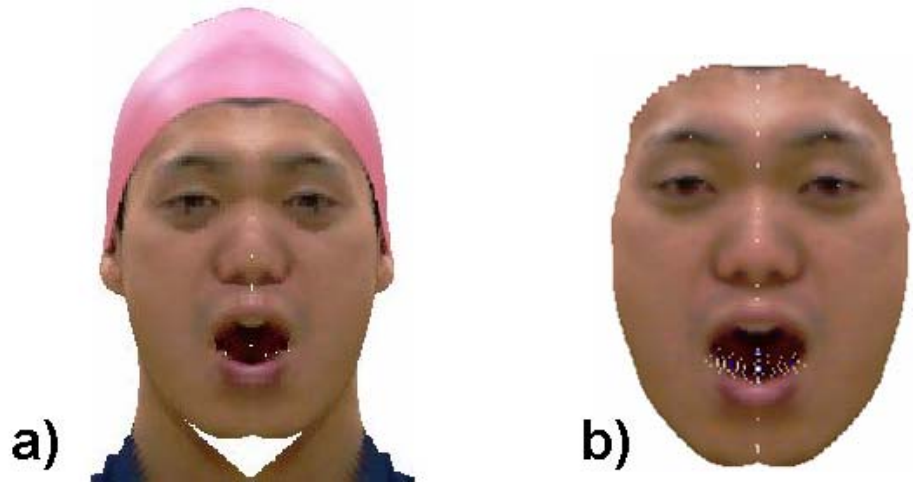

Figure 22. a) Reconstructed 3D face from surprise before segmentation. b) Facial surface after segmentation.

According to the pilot test, the average a* and b* value of the cap, facial surface and background are defined as 159.547, 131.251, 142.188, 152.240 and 128.000, 128.000

respectively. The reconstructed 3D face after segmentation is shown in Fig. 22.

3.1.2. Normalized Vector Computation

As normalized vectors pointing to the center of the head are used to compute the face plane, four neighboring points are used to compute a normalized vector pointing from a point on the facial surface. The four neighboring points of a point on the facial surface is shown in Fig. 23. A normal vector is computed by cross product according to Eq. 12.

$$\vec{c} = \vec{a} \times \vec{b} \tag{12}$$

where $\vec{a}$ and $\vec{b}$ are vectors pointing from a point on the facial surface to the neighboring points. $\vec{c}$ is a vector computed by cross product.

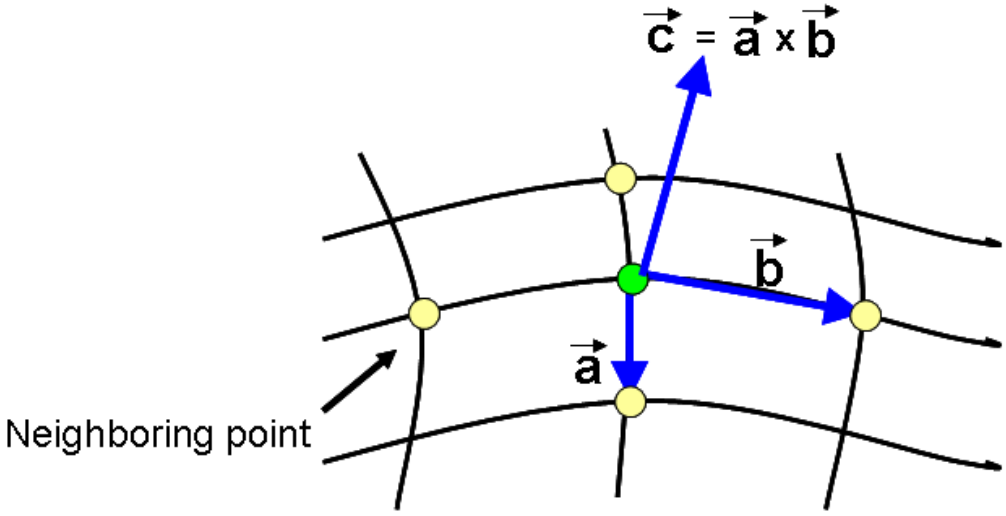

Figure 23. Four neighboring point and a normal vector

A normalized vector is the average of the four normal vectors computed from neighboring points. In case of a neighboring point including with noise data, the normal vector computed from data with noise always point to different direction from the others. According to this empirical property, the angles between each vector should be smaller than 30º. The normal vector, of which the angle is bigger than 30º from the other vectors, will be ignored. The remains are represented by a normalized vector. The normalized vector pointing from a point on the facial surface is shown in Fig. 24.

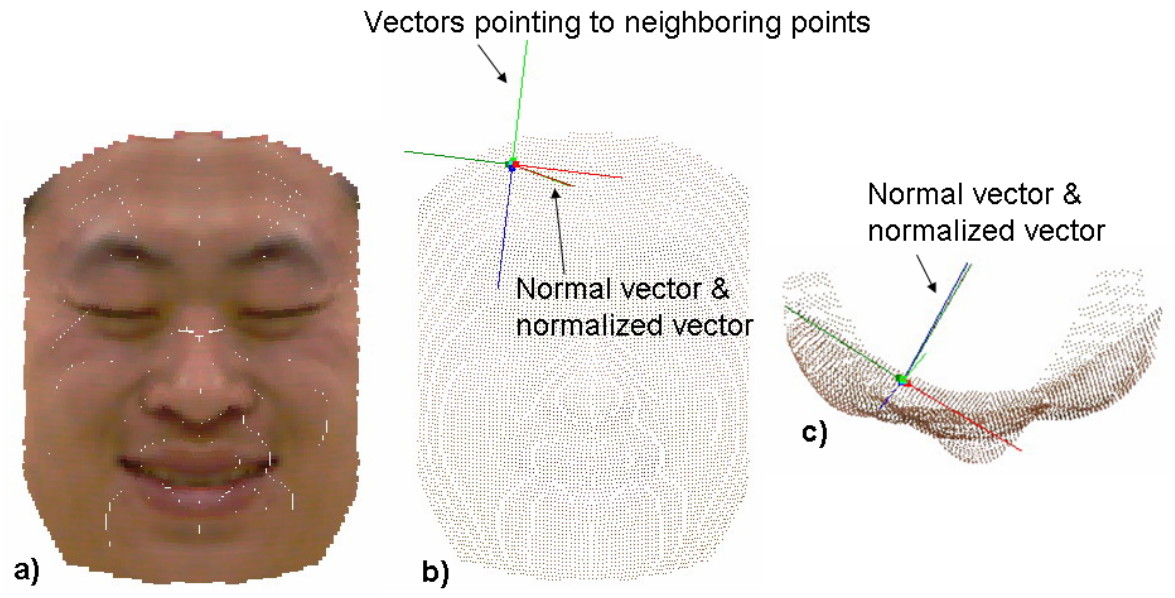

Figure 24. a) Smiling face. b) Vectors pointing to neighboring points, normal vectors and normalized vector from the frontal view of face. c) The vectors from the top view of face.

3.1.3. Derivation of the Face Plane

The face plane is a virtual plane across a head. A face plane is derived by normalized vectors as described in 3.1.2. The derivation of the face plane is described as follows.

The three dimensional data on the facial surface is described as $p_i = (a_i, b_i, c_i)^t, (i = 1, ..., N)$ (see Fig.25). Let the normalized vector on the normal line passes through a point p_i on the face be $f_i = (j_i, k_i, l_i)^t$. Now suppose a point P inside the head and orthogonal projection from the point P to the normal line. The distance d_i of orthogonal projection from the point to the normal line f_i is as follows:

$$d_i^2 = \|P - p_i\|^2 - (P - p_i, f_i)^2$$

$$= (P - p_i)^t (E - f_i \cdot f_i^t)(P - p_i) \tag{13}$$

where E is the unit matrix and $(\bullet, \bullet)$ is the inner product. The optimization can be carried out by differentiating $Q = \sum_i d_i^2$ with respect to P.

$$\frac{dQ}{dP} = 2\sum_i (E - f_i \cdot f_i^t)(P - p_i) = 0 \tag{14}$$

Then the optimal point P can be obtained.

$$P = \left[\sum_i (E - f_i \cdot f_i^t) \right]^{-1} \cdot \sum_j (E - f_j \cdot f_j^t) p_j \tag{15}$$

The face plane passes through this point. But the orientation of the plane has not been determined. Therefore, the direction n which makes the next equation maximum is solved. This makes an eigen problem under the condition $\|n\| = 1$.

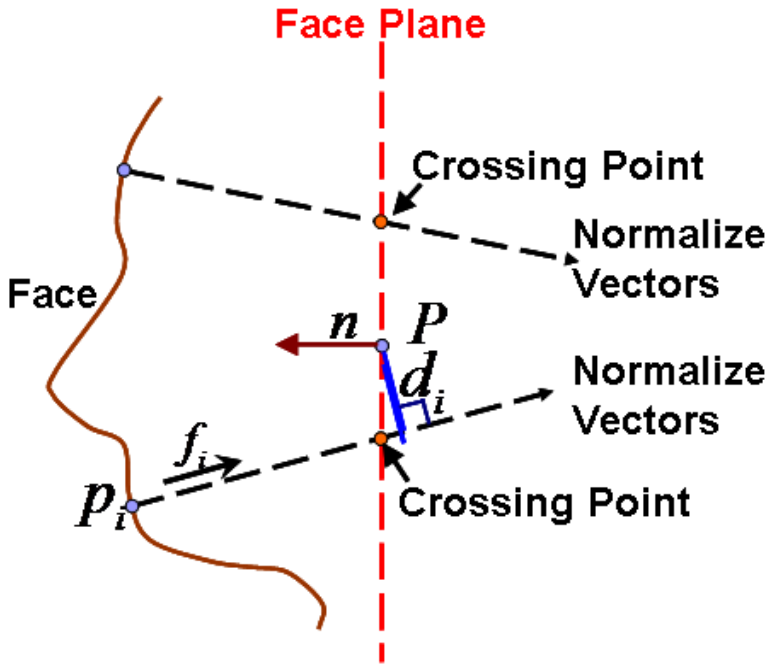

Figure 25. Derivation of face plane

$$S = \sum_i (f_i, n)^2 = n^t \left(\sum_i f_i \cdot f_i^{\,f} \right) n \tag{16}$$

By solving this problem, the three eigenvalues and three corresponding eigenvectors can be obtained as $\lambda_1 \geq \lambda_2 \geq \lambda_3$ and ϕ_1, ϕ_2, ϕ_3 respectively. ϕ_1 will be chosen as the direction n of the face plane based on the least distance. The face plane is fixed by P and n.

3.1.4. Crossing Point Computation

Based on the face plane computed in 3.1.3., the crossing points can be calculated. As

the normalized vector $f_i = (j_i, k_i, l_i)^t$ points from the point $p_i = (a_i, b_i, c_i)^t$, the equation of the normal line is described as follows;

$$h_i = p_i + t_i f_i \tag{17}$$

where $h = (x, y, z)^t$ is a point on the normal line, t_i is a real number. Describing Eq. (17) in element-wise, Eq. (18) can be obtained. (see Fig. 26)

$$\begin{aligned} x_i &= a_i + t_i j_i \\ y_i &= b_i + t_i k_i \\ z_i &= c_i + t_i l_i \end{aligned} \tag{18}$$

Then, suppose a virtual plane in a head;

$$(n, P - X_i) = 0 \tag{19}$$

where $n = (A, B, C)^t$ is the normal vector of the face plane, $X_i = (x_i, y_i, z_i)^t$ and $P = (x_p, y_p, z_p)^t$ are points on the face plane. Substituting Eq. (18) in Eq. (19), t_i can be obtained. By substituting t_i in Eq. (18), the crossing point X_i can be obtained (see Fig. 1.(d)).

$$t_i = \frac{(Aa_i + Bb_i + Cc_i) - (Ax_p + By_p + Cz_p)}{(Aj_i + Bk_i + Cl_i)} \tag{20}$$

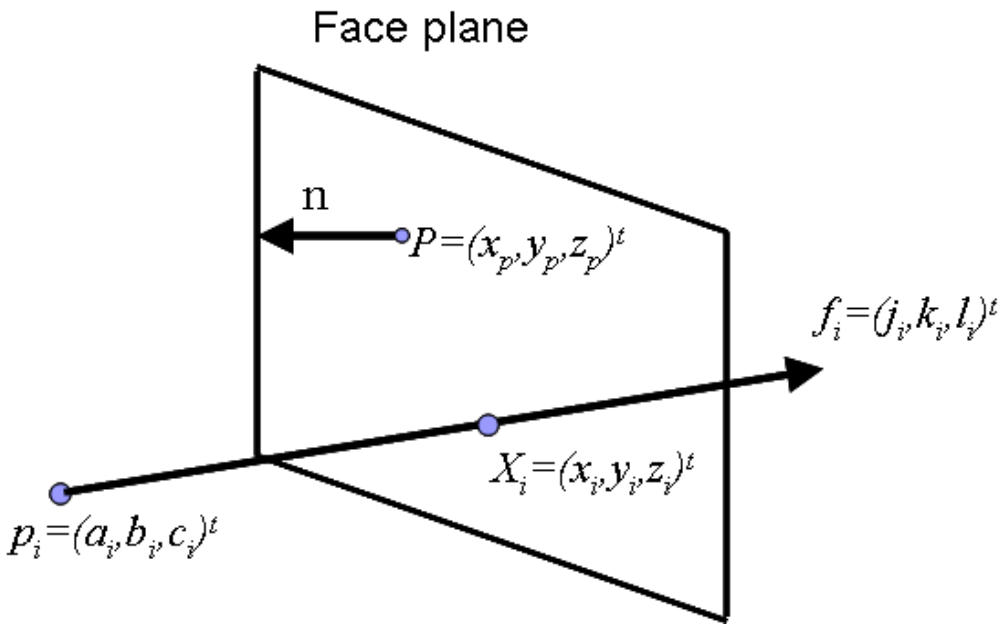

Figure 26. The normalized vector f_i points from a point p_i on the facial surface. X_i is a crossing point on the face plane.

3.2. Crossing Point Analysis Schemes

As the facial expression is represented in terms of the crossing points, the crossing points are analyzed by using two schemes for recognition: crossing point distribution scheme and displacement vector scheme.

3.2.1. Crossing Point Distribution

In this subsection, the crossing point distribution is considered. The face plane is divided into 36(6x6) sub-areas (see Fig. 27). A set of crossing points in each sub-area is used as pattern feature.

To define the 36 sub-areas, the center C is defined as the average of crossing points (see Fig. 27). The standard deviation on x-axis (σ_x) and y-axis (σ_y) of the crossing points within the area of 20 cm. from the center C are calculated. The width and height of the region of interest (ROI) are defined as $2*1.8*\sigma_x$ and $2*1.8*\sigma_y$ respectively, where the factor 1.8 is determined by a pilot test. The ROI is divided into 36(6x6) sub-areas. The crossing points in each sub-area are counted and normalized in Eq. (21).

$$
\begin{aligned}
W &= (w_1, w_2, w_3, ..., w_n)^t \\
\overline{w} &= \frac{1}{n}\sum_{a=1}^{n} w_a \\
V_c &= \left(w_1 - \overline{w}, w_2 - \overline{w}, w_3 - \overline{w}, ..., w_n - \overline{w}\right)^t \\
\mathbf{V} &= \left(\frac{w_1 - \overline{w}}{\|V_c\|}, \frac{w_2 - \overline{w}}{\|V_c\|}, \frac{w_3 - \overline{w}}{\|V_c\|}, ..., \frac{w_n - \overline{w}}{\|V_c\|} \right)^t
\end{aligned}
\tag{21}
$$

where w_a is the number of crossing points in each sub-area. a is the sub-area number starting from 1 to 36. $\mathbf{V}$ is the normalized vector of the feature vector V_c and has 36 components.

For recognition, the facial expression change from neutral face is considered. Therefore, each facial expression is compared with the referential neutral face of the same person (see Eq. (22)). Data used for recognition is:

$$
\mathbf{Y}_\alpha^{(k)} = \mathbf{V}_\alpha^{(k)} - \mathbf{V}_\alpha^{n} \tag{22}
$$

where $\mathbf{Y}_\alpha^{(k)}$ is the difference vector. $\mathbf{V}_\alpha^{(k)}$ is the normalized feature vector of facial

expression. $\mathbf{V}^n$ is the normalized feature vector of reference neutral face. α is the data number. k is the category number of facial expression.

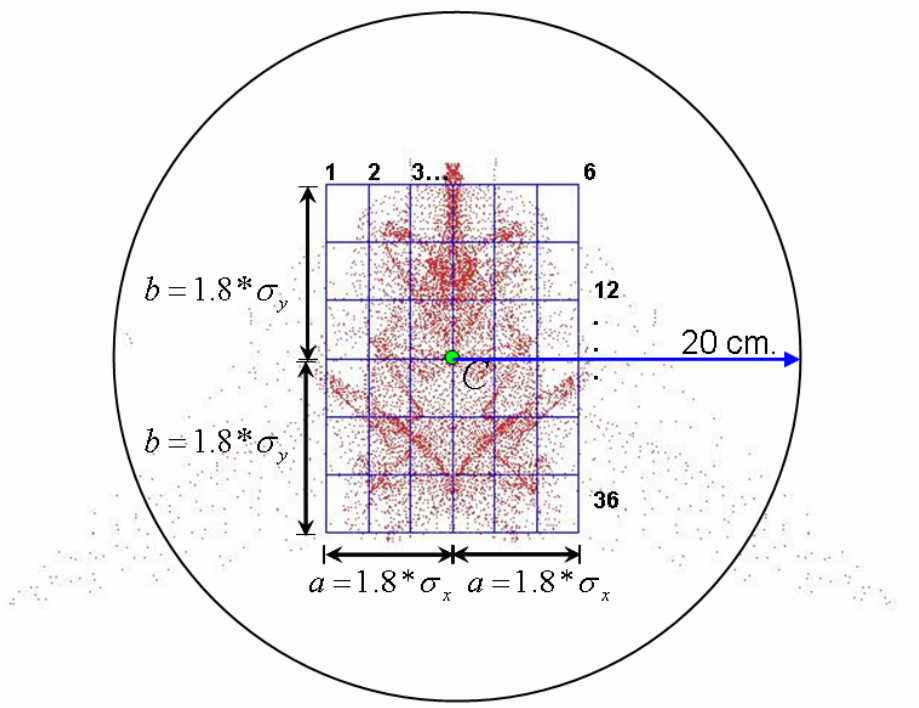

Figure 27. The crossing point distributions in the 36(6x6) sub-areas

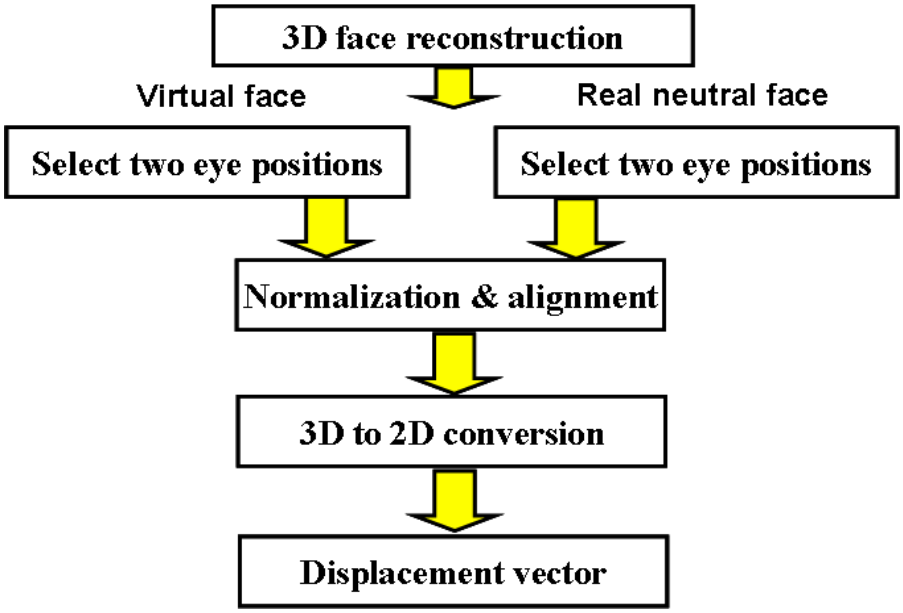

Figure 28. A flowchart of the displacement vector method algorithm

3.2.2. Displacement Vector

Based on the assumption that the person is known, each facial expression image is compared with the neutral face of the same person. The size of the neutral face and the expression face has to be the same to compute the displacement vector. The two ear positions of both faces are selected manually for the normalization, and then the expression face is normalized to the neutral face. The 3D faces are converted to 2D faces from the frontal view face. Pair of points on the both face surfaces which have the same xy-coordinate of the neutral face and the expression face are considered. The pair of crossing points on the face plane corresponding to the pair of points on the two face surfaces will be used for computing a displacement vector. A flowchart of displacement vector method algorithm is shown in Fig. 28. In this subsection, the expression face size normalization, 3D to 2D conversion and displacement vector will be described.

A) Expression Face Size Normalization. Because the size of the neutral face and the expression face has to be the same for the displacement vector computation, the size of the expression face is normalized to be the same size of the neutral face by using two eye positions. The two eye positions are select manually as shown in Fig. 29. The normalization is described as follow:

$$s = \sqrt{\left(x_1 - x_2\right)^2 - \left(y_1 - y_2\right)^2} \tag{23}$$

$$Fc = \frac{s^n}{s^e} \tag{24}$$

$$\left[X^w - X^c\right] = Fc\left[X^e - X^c\right] \tag{25}$$

where s is the distance between two eyes, X is the 3D-coordinate, X^c is the average of the facial surface data, X^w is the new 3D-coordinate, e is the expression face, n is the neutral face and.

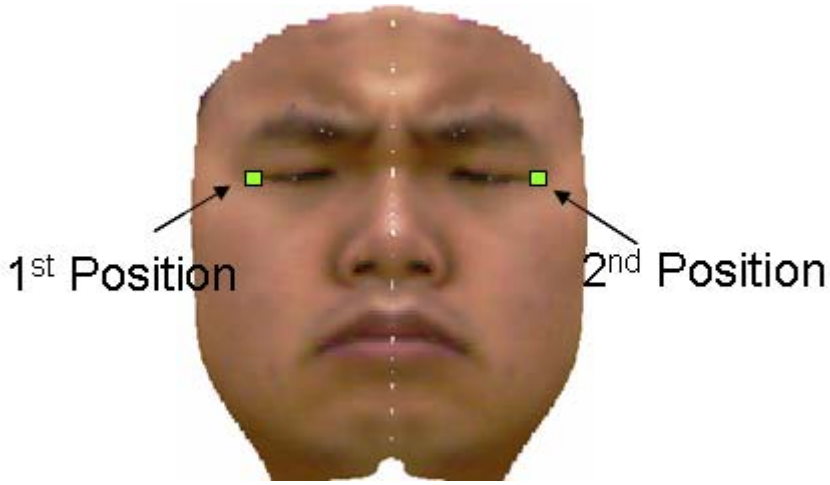

Figure 29. Two eye positions are selected manually. The green points are the two positions of the eyes.

B) 3D to 2D conversion. A scale used to convert 3D data $X_i^{3D} = \left(x_i^{3D}, y_i^{3D}, z_i^{3D}\right)$ to 2D data $X_i^{2D} = \left(x_i^{2D}, y_i^{2D}\right)$ is 1.5 mm. The nose tip point $X_i^n = \left(x_i^n, y_i^n, z_i^n\right)$ is defined as the reference point. 2D-coordinate is defined as follow:

$$x_i^{2D} = \begin{cases} \left\lfloor \dfrac{x_i^{3D} - x^n}{1.5} \right\rfloor & ; x_i^{3D} - x^n < 0 \\ \left\lceil \dfrac{x_i^{3D} - x^n}{1.5} \right\rceil & ; x_i^{3D} - x^n > 0 \end{cases}$$

$$y_i^{2D} = \begin{cases} \left\lfloor \dfrac{y_i^{3D} - y_i^n}{1.5} \right\rfloor & ; y_i^{3D} - y^n < 0 \\ \left\lceil \dfrac{y_i^{3D} - y_i^n}{1.5} \right\rceil & ; y_i^{3D} - y^n > 0 \end{cases}$$

C) Displacement Vector. A displacement vector is a movement vector of the crossing point pointing from the crossing point of neutral face to the one of expression face. The pair points on the surface of neutral face and expression face which have the same xy-coordinate are described as p_i^n and p_i^e, (i = 1,…,N), respectively. The two crossing points c_i^n and c_i^e are the positions on the face plane corresponding to p_i^n and p_i^e respectively. The displacement vector d_i^e is a vector pointing from c_i^n to c_i^e (see Fig. 30).

The surface of the expression face is divided into 36 (6x6) areas based on the size of

neutral face (see Fig. 1.(f)). Displacement vectors in each area are represented by an average vector as shown in Fig. 31. The displacement vector set V is consisted of 36 average displacement vectors v_m.

$$d_i^e = c_i^e - c_i^n \tag{13}$$

$$v_m^e = \frac{1}{s}\sum_{i=1}^{s} d_i^e \tag{14}$$

$$V^e = \left(v_1^e, \ldots, v_m^e\right) \tag{15}$$

where c_i is the crossing point, d_i is the displacement vector, v_m is the average of displacement vector for points p_i^e on the surface in area m, (m = 1,…,36), s is the number of crossing points in the area m, e is the category of expression face and n represents the neutral face.

The displacement vector V^e is used for the facial expression recognition by means of a support vector machines method.

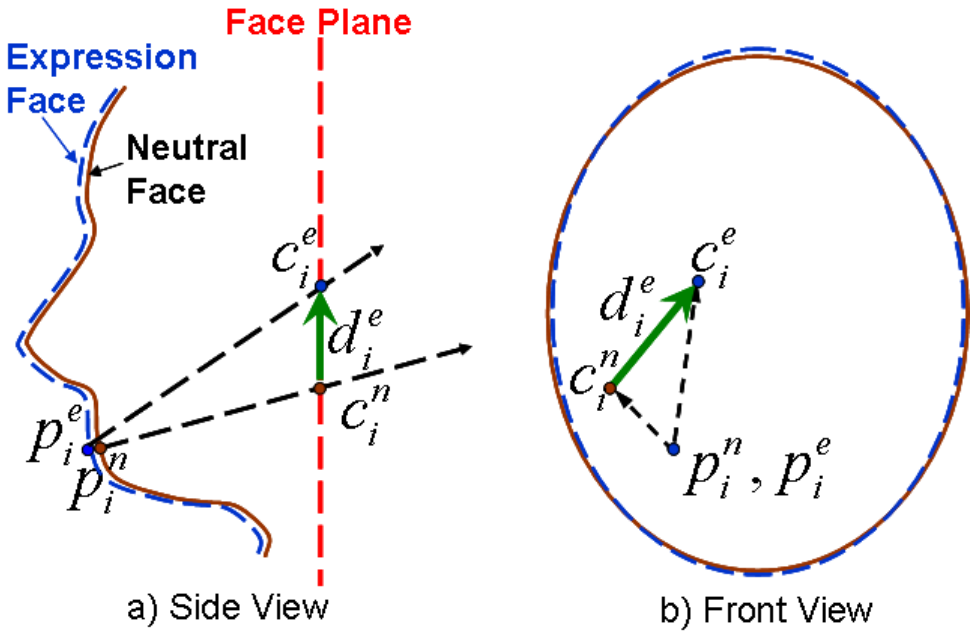

Figure 30. The concept of displacement vector

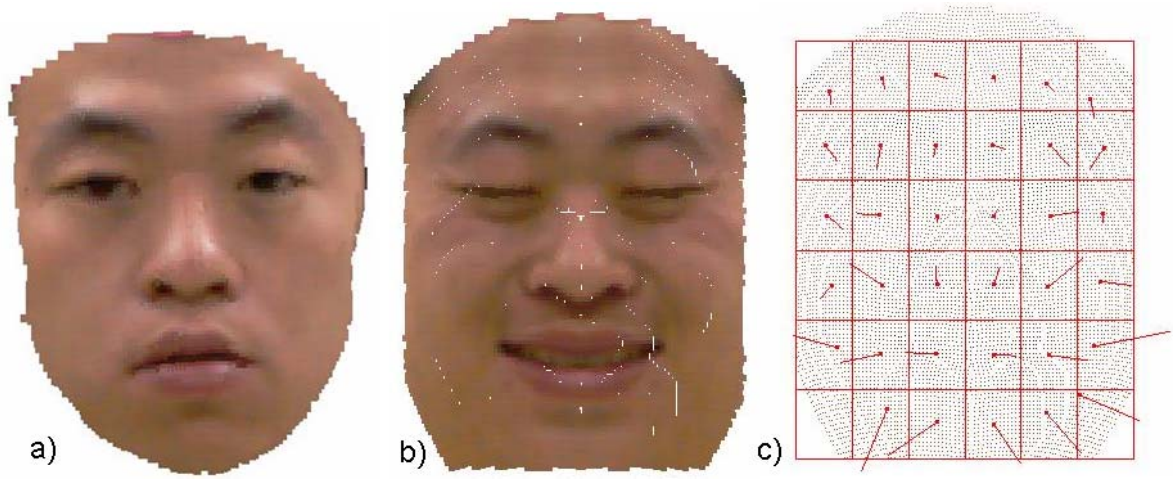

Figure 31. a) Real frontal face. b) Smiling face. c) Displacement vector set of smiling face.

Chapter 4 Experimental Results

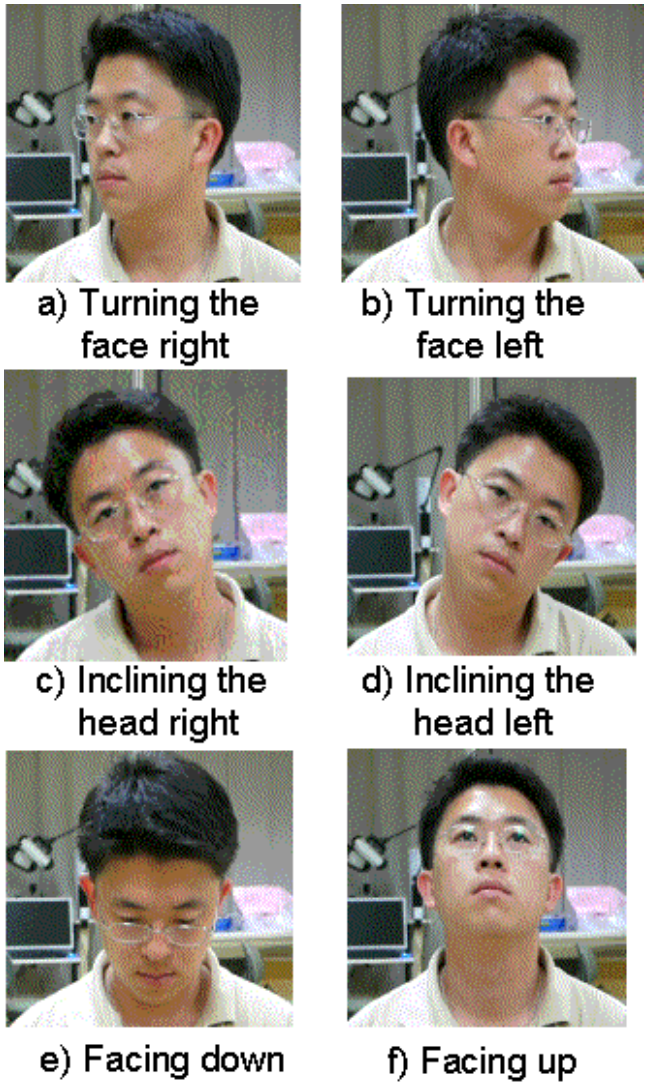

Figure 32. First row is head pose variation in yaw. Second row is head pose variation in pitch. Third row is head pose variation in roll.

To test the performance of the proposed method, the following three experimental results are considered: the nose ridge detection, 3D face reconstruction and facial expression recognition. All the experiments were implemented in MATLAB compiler version R2006a. In the experiments of the nose ridge detection and 3D face reconstruction, two conditions – one with the cap and one without the cap, were considered. The main variations of the head pose were the following: 1) turning the head left or right (yaw: rotation around the y-axis), 2) with the head facing up or down (pitch: rotation around the x-axis) 3) inclining the head (roll: rotation around the z-axis) (see Fig. 32). Our system is

tested for yaw variation in sections 4.1 and 4.2. In the experiment of the facial expression recognition, the condition with the cap is considered. The successful data of the 3D face reconstruction are used for the facial expression recognition in the experiment in section 4.3. The support vector machines[40] with the radial basis function is used for recognition by Leave-One-Out method. The results of the 3D face reconstruction for pitch and roll in the case of the front face will be considered in section 4.4.

4.1. Nose Ridge Detection

The results of the yaw variation from 9 different viewpoints ranging from -80 degrees to +80 degrees are shown in Table 1. Failure occurred in our experiment when other parts of a face such as an ear or a cheek were detected. The upper row in Table 1 shows the case with the cap, the bottom row is the case without the cap. As mentioned in subsection 2.1, the range scanner is insensitive to black colors such as typical Asian hair. Therefore, some people changed their black hair by applying white powder during the experiment.

Table 1. Experimental results of the nose ridge detection by turning the face to left and right (Yaw)

Nose ridge detection		Yaw									
		Rotation angle around the Y axis (degree)									Total
		-80	-67.5	-45	-22.5	0	22.5	45	67.5	80	
With a cap	Total number of samples	17	27	27	27	27	27	27	27	17	223
	Number of failed samples	0	0	0	0	0	0	0	0	0	**0**
	Nose ridge detection ratio (%)	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100**	**100.0**
Without a cap	Total number of samples	12	12	12	12	12	12	12	12	12	108
	Number of failed samples	0	**1**	**2**	**1**	**1**	**1**	**3**	**1**	0	**10**
	Nose ridge detection ratio (%)	**100**	**91.7**	**83.3**	**91.7**	**91.7**	**91.7**	**75.0**	**91.7**	**100**	**90.7**

The total number of test samples was 223 with the cap and 108 without the cap. The nose ridge of all samples with the cap was detected successfully from the results with the cap. However, without the cap, the total nose ridge detection ratio obtained was 90.7%. Ten samples were unsuccessful, obviously because of the influence of messy hair.

To evaluate the results, the real nose tip and the ideal symmetry plane were determined manually from 321 successful samples. The accuracy of the nose tip detection

was smaller than 1.0 mm. The average of the absolute value of error was 0.5 mm. The standard deviation was 0.5 mm. The inclination of the symmetry plane from the ideal symmetry plane was less than 2.1 degrees. The average of the absolute value of the inclination error was 1.2 degrees and the standard deviation was 0.5 degrees.

The examples with the cap and without the cap with yaw variation are shown in Fig. 33 and 34, respectively. The nose location is illustrated by the 15 points starting from the nose tip.

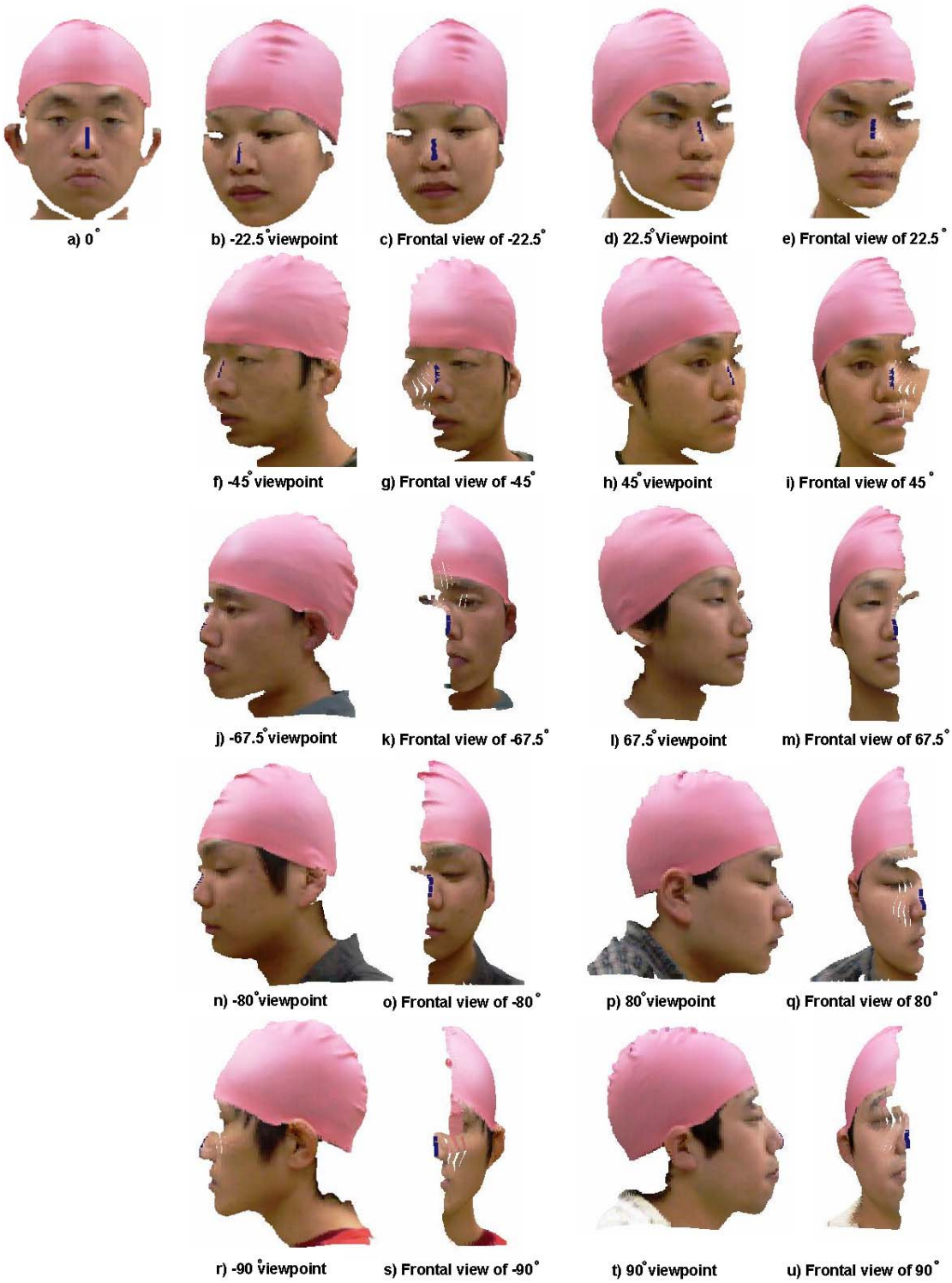

Figure. 33. Visual results of the nose ridge detection with the face with the cap captured from -80 degrees to +80 degrees in yaw. The side viewpoint (a, c, e, g, i, k, m, o, q, s, u) is shown with its front viewpoint (b, d, f, h, j, l, n, p, r, t) respectively. The points on the nose are the result of the nose ridge and tip detection.

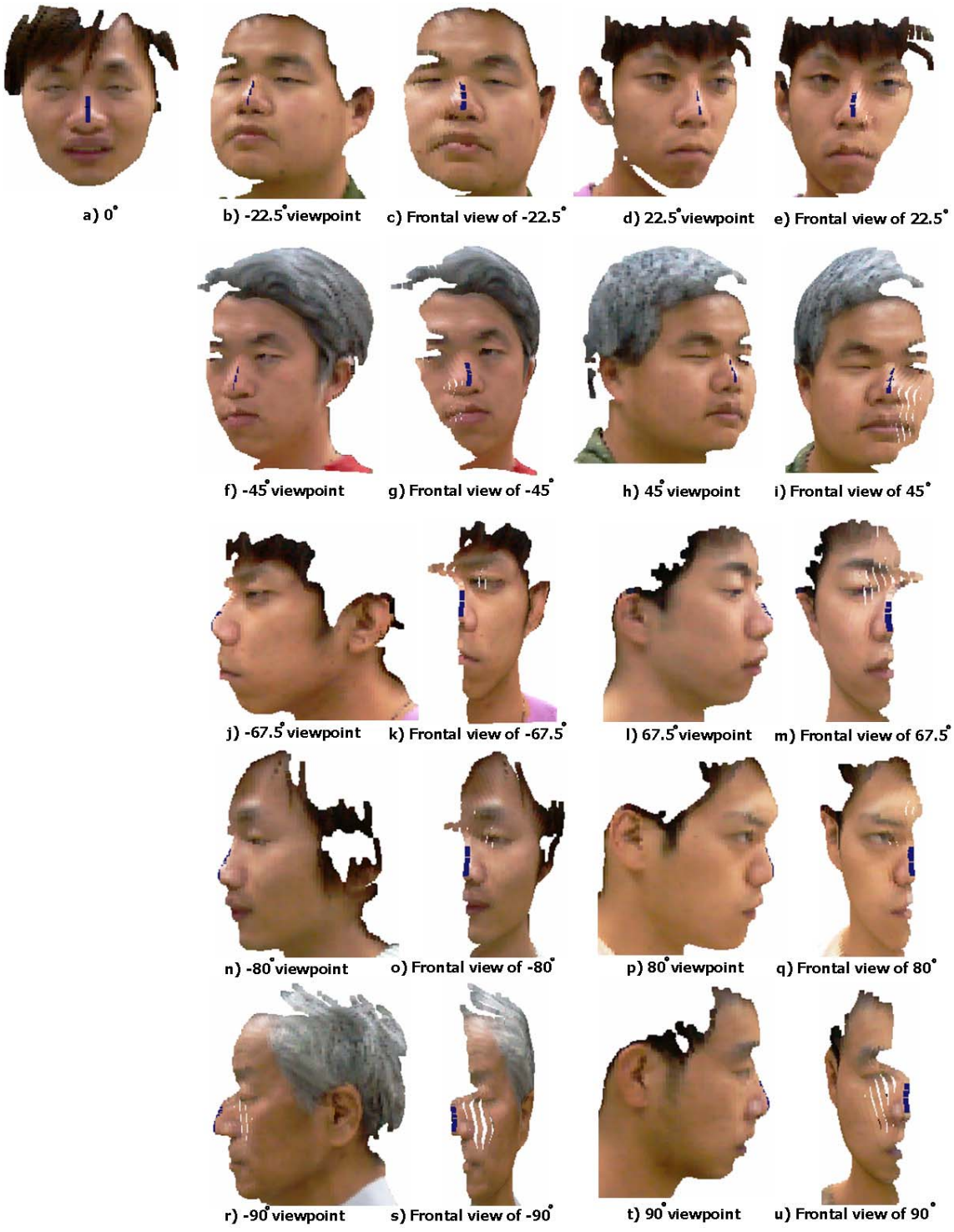

Figure. 34. Visual results of the nose ridge detection with the face without the cap captured from -80 degrees to +80 degrees in yaw. The side viewpoint (a, c, e, g, i, k, m, o, q, s, u) is shown with its front viewpoint (b, d, f, h, j, l, n, p, r, t) respectively. The points on the nose are the result of the nose ridge and tip detection.

4.2. 3D Face Reconstruction

In this experiment, the head poses in yaw were reconstructed from both situations – with the cap and without the cap. The virtual faces were evaluated using a criterion obtained from a facial recognition test according a flowchart in Fig. 35. To do this test, the virtual face was aligned to the frontal real face (frontal view) according to six degrees of freedom (x, y, z, roll, pitch and yaw). The nose tip points and the nose ridge vectors of two faces were used as the local points to match the surfaces of two faces. The errors between two faces were minimized by a Nelder-Mead simplex method. The matching distance was then computed (see Fig. 36).

In the recognition test, the virtual face was matched with the frontal real faces of 27 persons. When the person with the smallest matching result is the same person, the recognition result was considered correct. Then, four virtual faces were recognized incorrectly as shown in Table 2.

To determine the criterion, the matching distance and the standard deviation of the correct facial recognition results are shown in Fig. 37. The matching distances of the incorrect recognition were larger than 2 mm. Therefore, 2 mm was set as the criterion for evaluating the virtual faces.

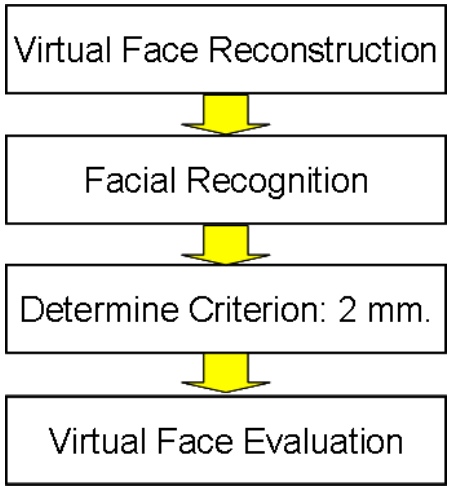

Figure 35. Flowchart of reconstructed 3D face evaluation algorithm

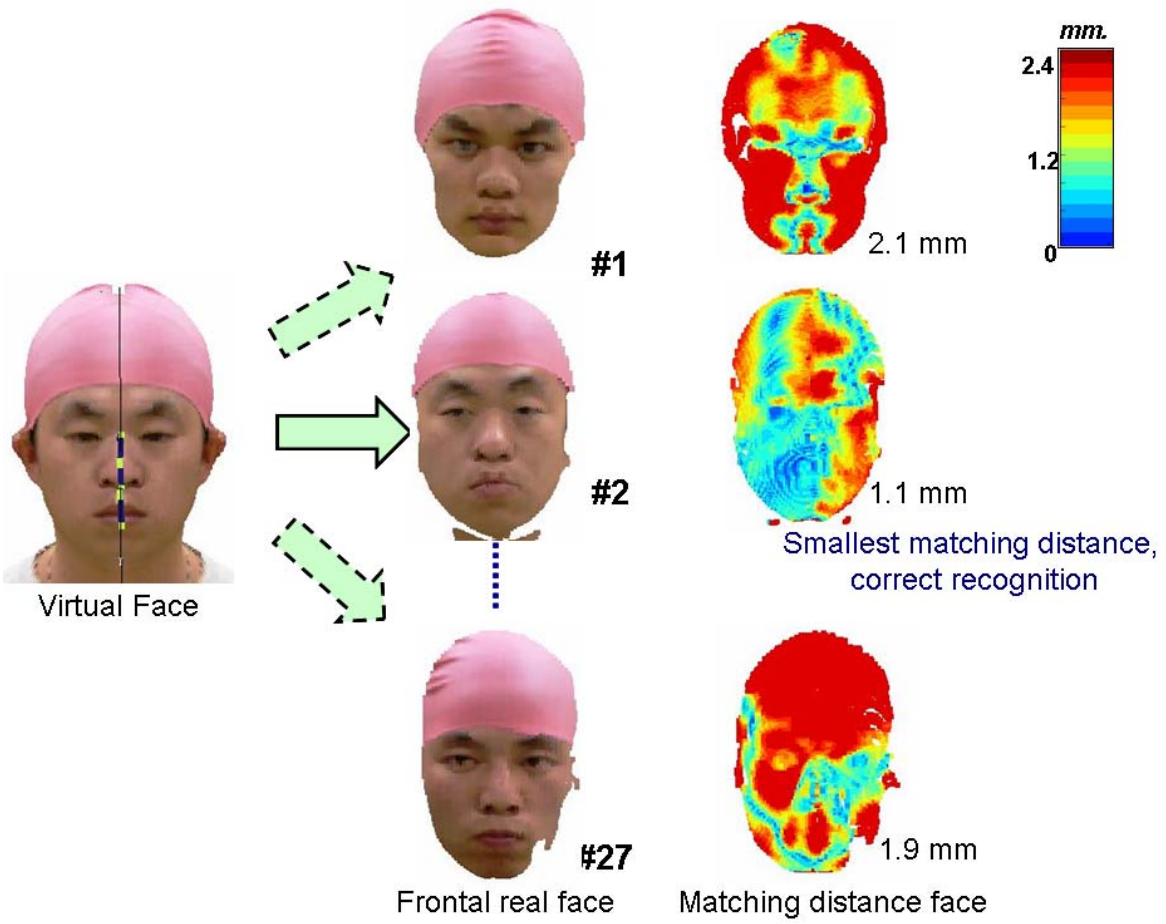

Figure. 36. Each virtual face was matched with 27 frontal real faces. The matching distance was then computed. Matching distance of each pixel is shown in different colors. Red color is the long matching distance. The blue color is the shortest matching distance. When the person with the smallest matching result is the same person, the recognition result was considered correct.

Table 2. Recognition accuracy by matching the virtual face with the frontal real face

Viewpoint of face (Degree)	Total number of sample	Number of failed samples	Recognition accuracy
-45	27	4	85.2%
-22.5	27	0	100.0%
0	27	0	100.0%
22.5	27	0	100.0%
45	27	0	100.0%
Total	135	4	97.0%

Using this criterion, the virtual faces with shorter matching distances than 2 mm were judged to be a correct facial reconstruction. While the virtual faces with longer matching distances than the criterion were judged as an incorrect facial reconstruction. The data captured from different viewpoints ranging from -45 degrees to 45 degrees in Table 1 were used to reconstruct the virtual faces. There were 135 data sets from 27 people with the cap and 60 data sets from 12 people without the cap. Eight data sets represented false nose ridge detections. Therefore, 187 data sets were used for the facial reconstruction. By using the criterion, 183 of the reconstructed faces were evaluated to be a correct reconstruction but four of the reconstructed faces were evaluated as an incorrect reconstruction. The computational time for 3D face reconstruction was 16.1 seconds for one data set using an Intel Core(TM) 2 Duo CPU 3.16 GHz processor with 3.25 GB of RAM.

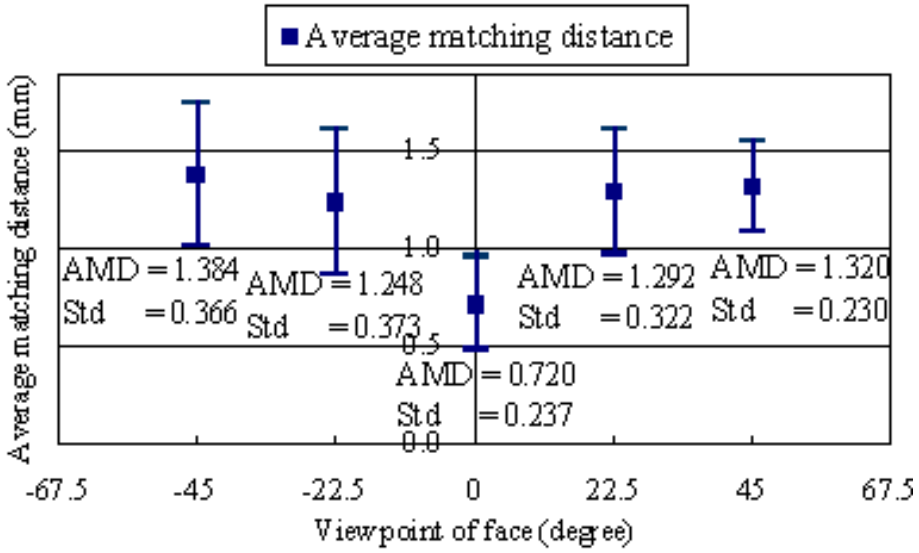

Figure. 37. Average matching distance (AMD) for all correct surface matching results and the standard deviation (Std) for each angle of view.

In Figs.38, 39, 42 and 43, we show results of the 3D face reconstruction without texture information so that readers can examine the results visually without any bias from the texture information. The virtual faces which were judged as a correct reconstruction were seen in Fig. 38. The virtual faces which were judged as an incorrect reconstruction were fatter than the real face of the same person as seen in Fig. 39. Examples of the virtual faces with and without a cap at various yaw angles are shown in Fig. 40 and 41 respectively. The virtual faces without texture information in Fig. 40 and 41 are also shown

in Fig. 42 and 43 respectively. The first row is a real face captured from data ranging from -45 degrees to 45 degrees. The second row is the virtual face. The third row is a three-quarter view of the reconstructed 3D face which represents the mirror face.

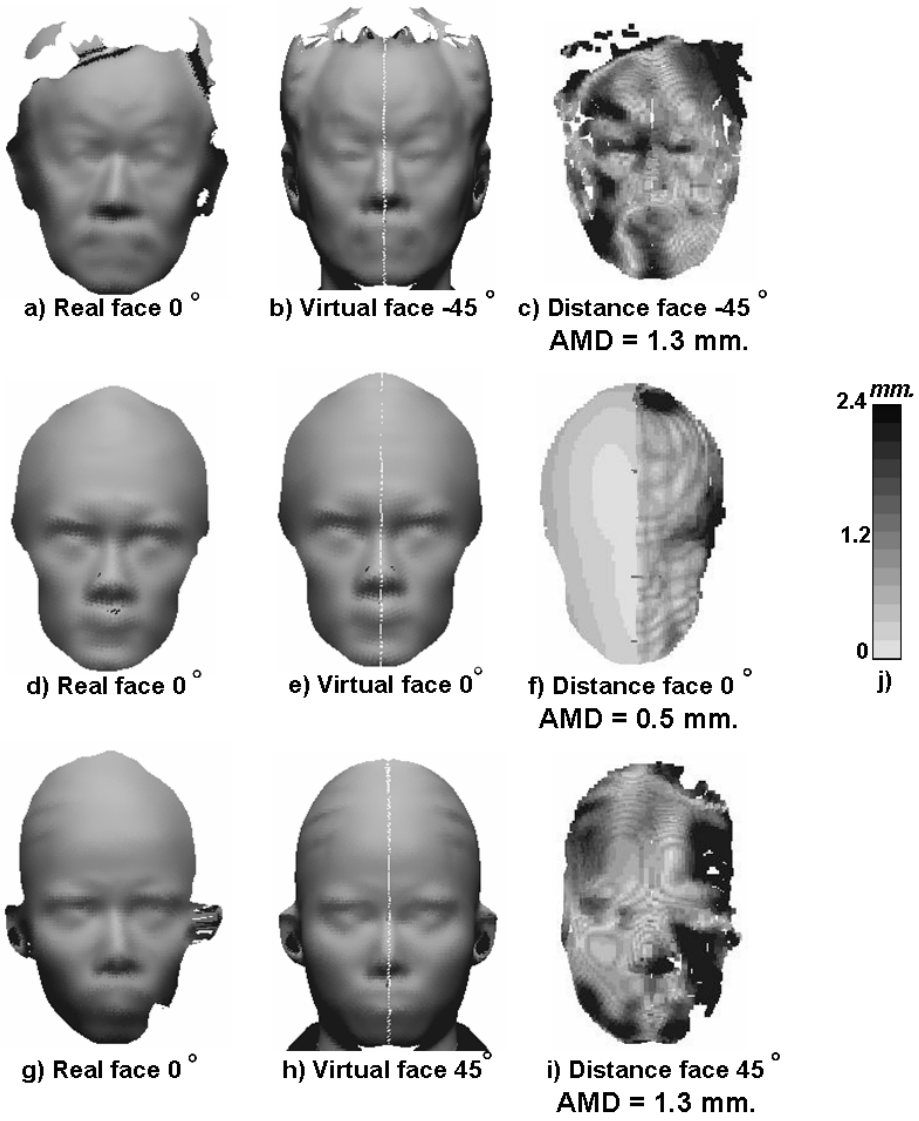

Figure. 38 The average matching distance between the frontal real face and the virtual face is shown by the gray colors. These virtual faces were judged to be a correct facial recognition by the criterion. The 3D faces in the first two columns are shown in terms of a polygon model without texture information.

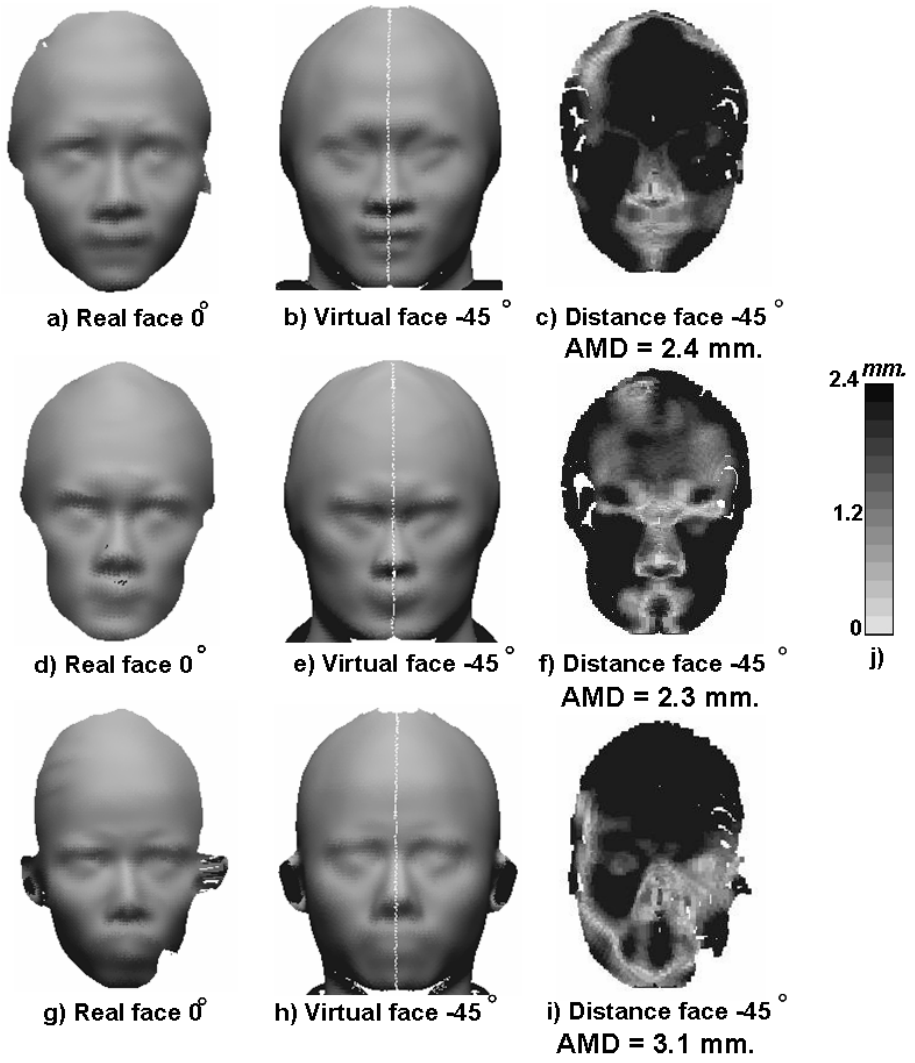

Figure. 39 The average matching distance of the virtual faces which were judged to be a incorrect facial recognition by the criterion.

The virtual faces without texture information showing the correct and incorrect reconstruction of the face are shown in Figs. 38 and 39 respectively. Each point on the virtual face matching the closest point on the frontal real face has a different distance. The matching distance of each point is shown by the different levels of gray according to a gray index (Fig.38.(s)). The virtual faces shown in the second column of Figs. 38 and 39 are matched with their frontal real faces shown in the first column of Figs. 38 and 39. In the third column of Figs. 38 and 39, the matching distance of each point on the virtual face is indicated by a different gray level. The matching distance of the virtual faces with an incorrect reconstruction (Fig. 39.(c,f,i)) shows a longer distance than the correct

reconstructed faces (Fig. 38.(c,f,i)). This problem occurs due to an error in estimating the average center point C_{Ae}.

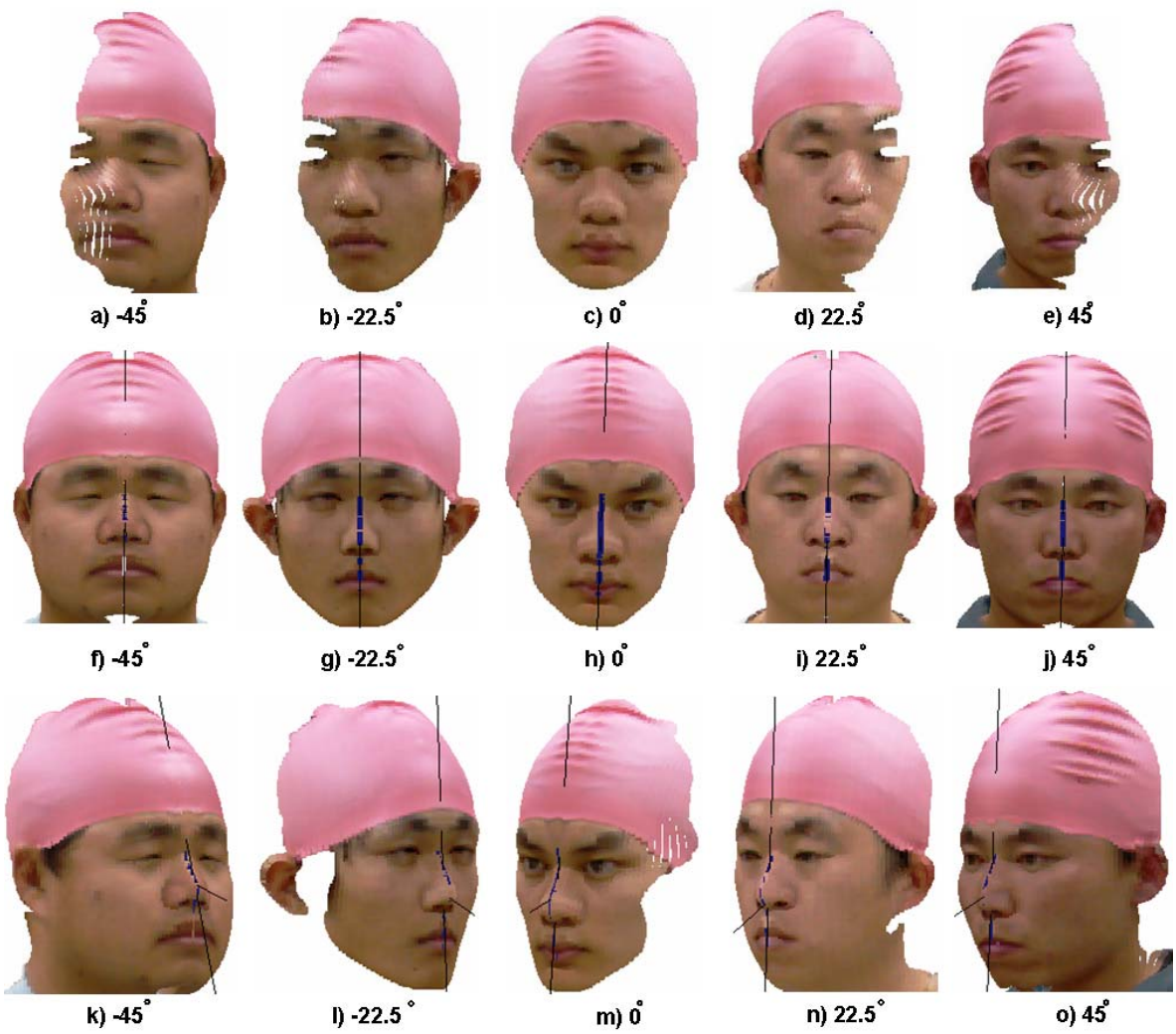

Figure. 40. 3D face reconstruction results with texture information from different viewpoint of yaw angle (turning left and right). The heads are cover by a cap. The first row is the real face data captured from different viewpoints from -45 degrees on (a). to +45 degrees on (e). The second row is the frontal view of a reconstructed 3D face. The center line of the face represents the symmetry plane. The third row is from the mirror face side of the virtual face.

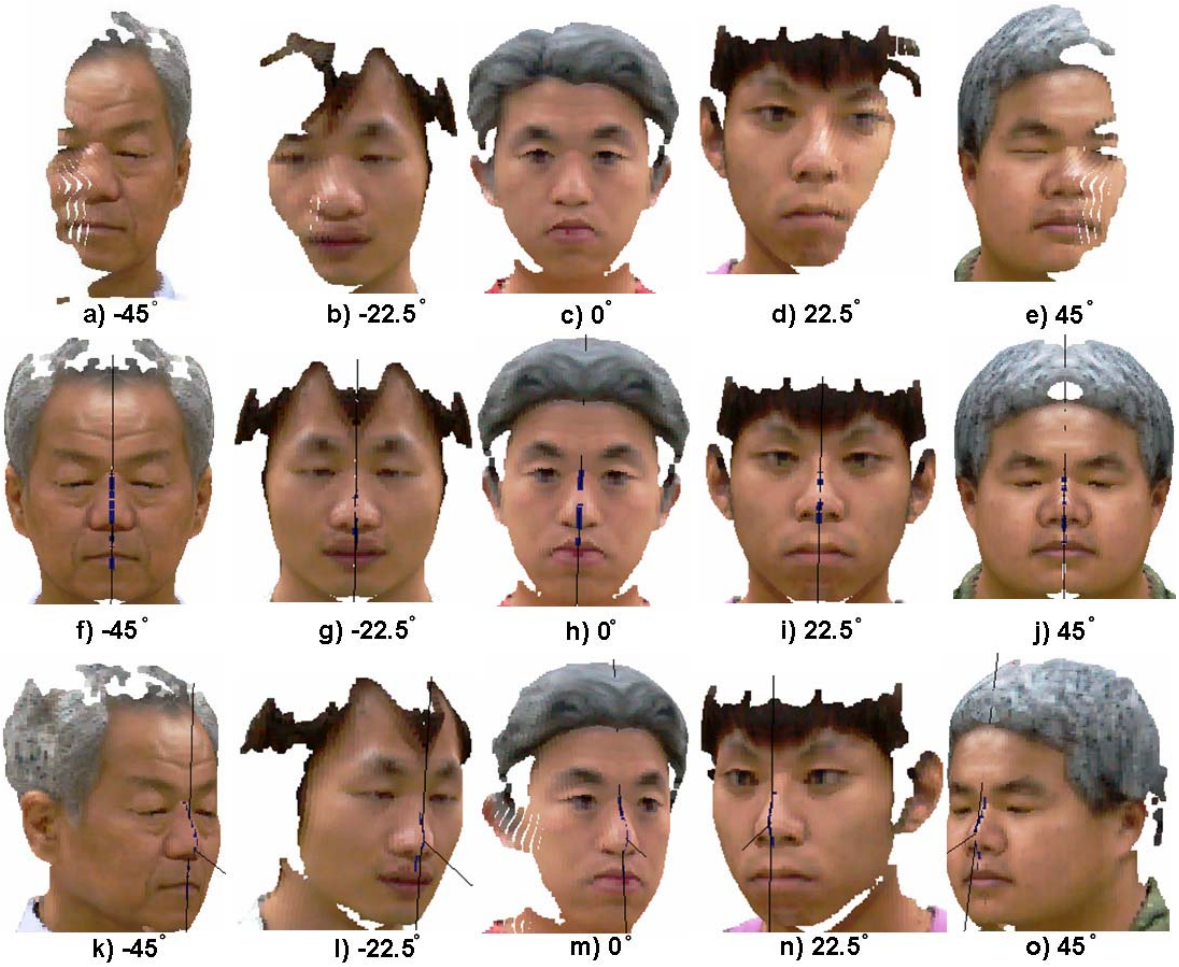

Figure. 41. 3D face reconstruction results without a cap.

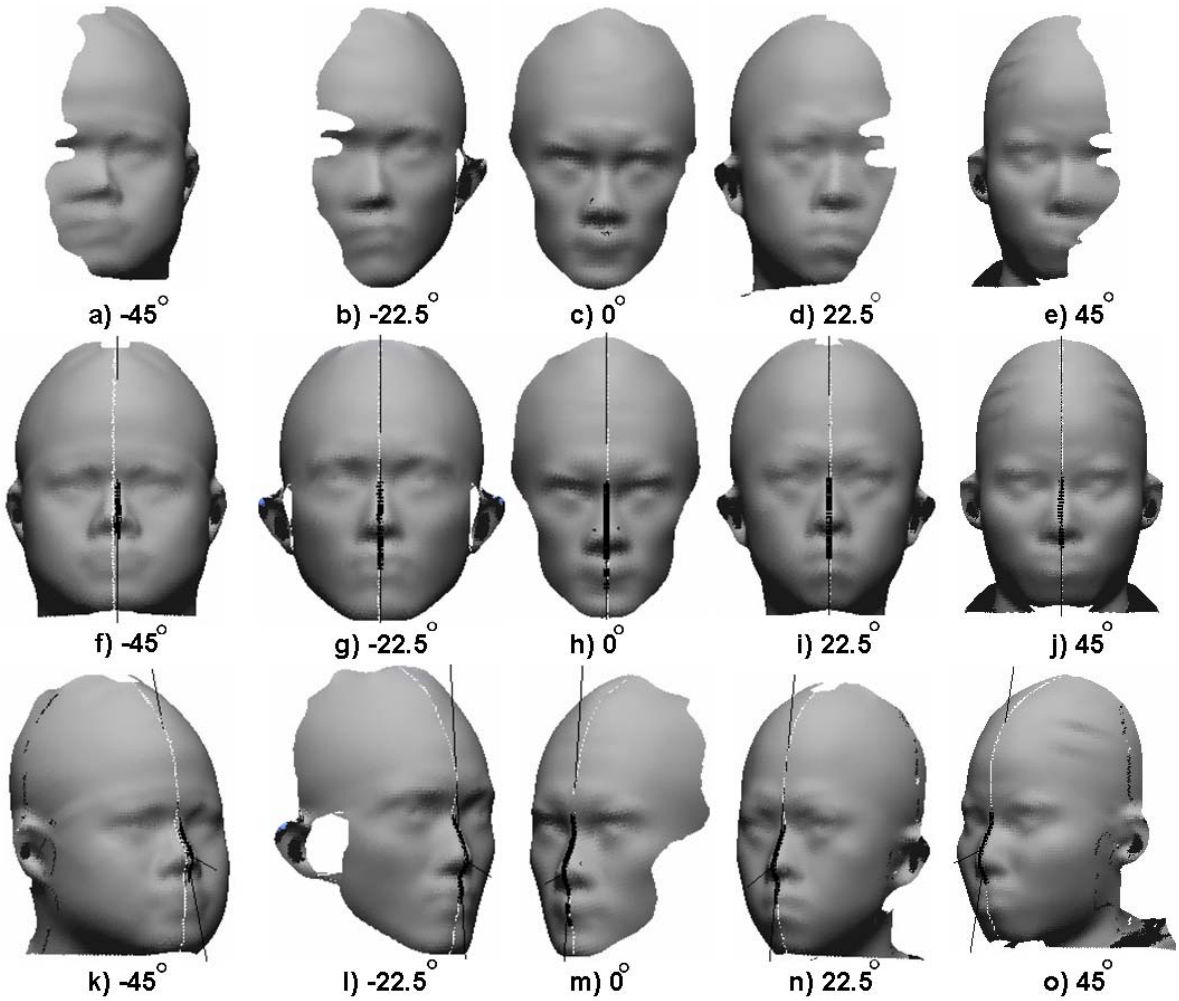

Figure. 42. The 3D virtual faces without texture information correspond to Fig. 40.

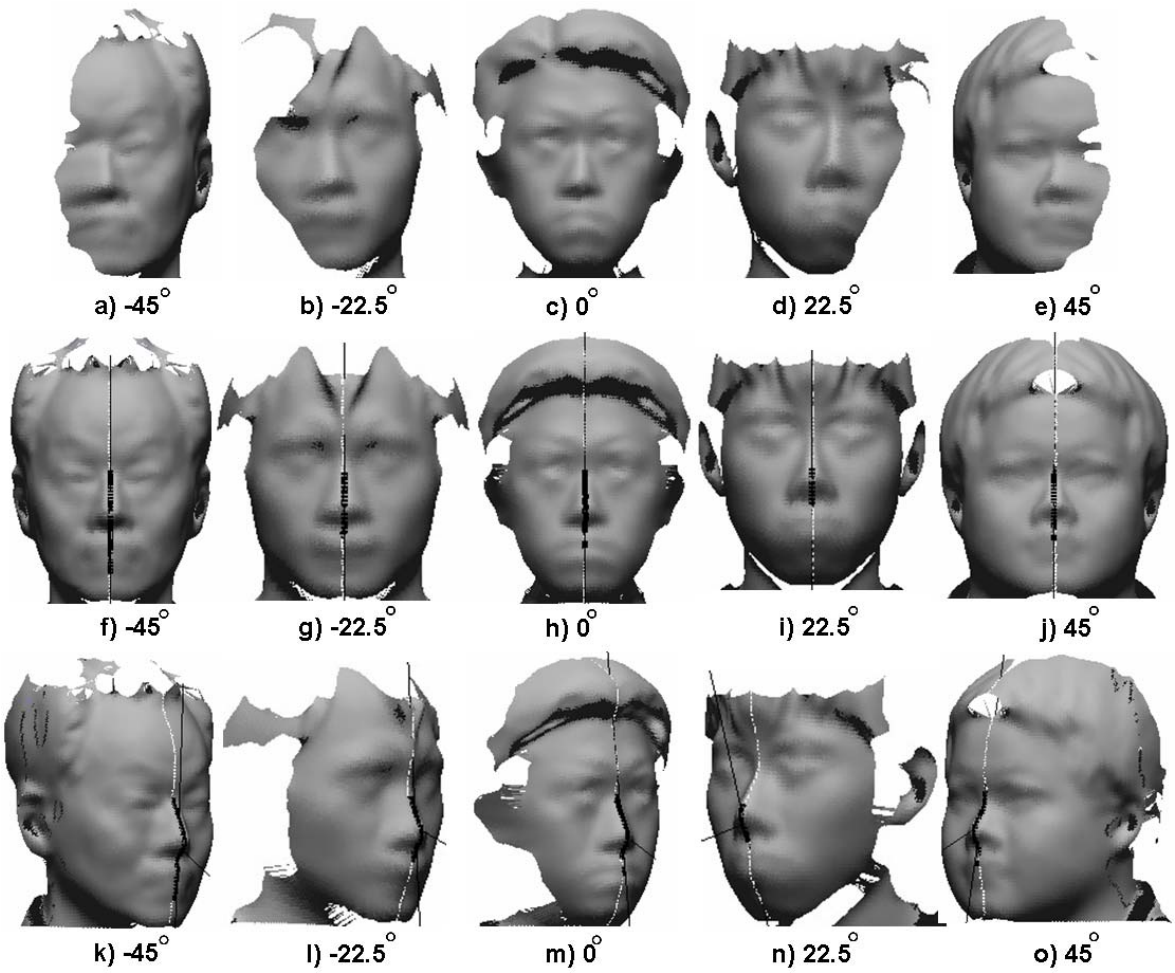

Figure. 43. The 3D virtual faces without texture information correspond to Fig. 41.

4.3. Facial Expression Recognition

In this section, there are two experiments for comparing the performance of the facial expression recognition. In subsection 4.3.1, the results for each expression of the face are compared. In subsection 4.3.1, the results for each viewpoint of the face are compared.

4.3.1. Comparison of Facial Expression

To test the performance of the crossing point analysis methods, the following two experimental conditions are considered: 1) crossing point distribution scheme, 2) displacement vector scheme. The successful reconstructed faces from 22 persons are used in this experiment. The experimental results of the crossing point distribution scheme and the displacement vector scheme of each facial expression and each viewpoint of facial data

are shown in Table 3. The recognition accuracy in Table 3 is also shown in Figs. 44 and 45.

In the experimental results, the accuracy of the smiling face is the highest because the characteristic of crossing point distribution is distinctive from the other. The accuracy of the anger face is the lowest because only partial surface around eyebrows changes from the neutral face. Therefore, there are incorrect recognitions between the anger faces and the neutral faces as shown in Figs. 46 and 47.

Action levels of the expression face also have the influence to recognition results. The characteristics of the crossing point distribution and displacement vector of an expression face from a person is different depending to the action levels. Fig. 48 is the crossing point distribution and displacement vector of a smiling face from a person in different action levels.

Furthermore, recognition accuracies for the training data of both schemes are also confirmed. The accuracies are 100%.

The computation time of the crossing point distribution method and the displacement vector method is 65.33 and 91.16 seconds, respectively, for a data set using Intel Core(TM) 2 Duo CPU 3.16 GHz processor with 3.25 GB of RAM.

Table 3. The facial expression recognition results by using the crossing point distribution scheme and the displacement vector scheme.

Expression face		Crossing point distribution method					Displacement vector method				
		−45°	−22.5°	0°	22.5°	45°	−45°	−22.5°	0°	22.5°	45°
Total number of sample	Anger	21	22	22	22	18	21	22	22	22	18
	Neutral	20	22	22	22	22	20	22	22	22	22
	Smiling	20	21	22	21	19	20	21	22	21	19
	Surprise	16	21	22	17	20	16	21	22	17	20
Number of correct recognition	Anger	8	8	13	8	8	13	13	14	16	13
	Neutral	10	8	11	15	13	12	18	19	18	16
	Smiling	15	19	20	16	13	19	20	22	20	17
	Surprise	7	15	21	14	16	9	17	19	13	17
Recognition accuracy (%)	Anger	38.1	36.4	59.1	36.4	44.4	61.9	59.1	63.6	72.7	72.2
	Neutral	50.0	36.4	50.0	68.2	59.1	60.0	81.8	86.4	81.8	72.7
	Smiling	75.0	90.5	90.9	76.2	68.4	95.0	95.2	100.0	95.2	89.5
	Surprise	43.8	71.4	95.5	82.4	80.0	56.3	81.0	86.4	76.5	85.0
Average accuracy		51.95	58.14	73.86	64.63	63.29	68.83	79.07	84.09	81.71	79.75

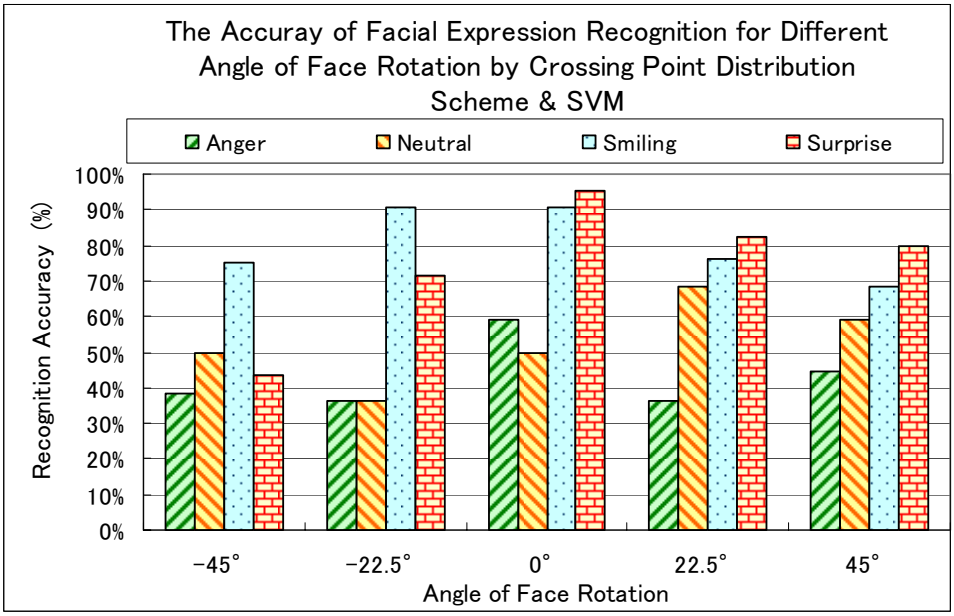

Figure 44. The accuracy of facial expression recognition for different face direction of -45°, -22.5°, 0°, +22.5° and +45° by using crossing point distribution scheme and SVM.

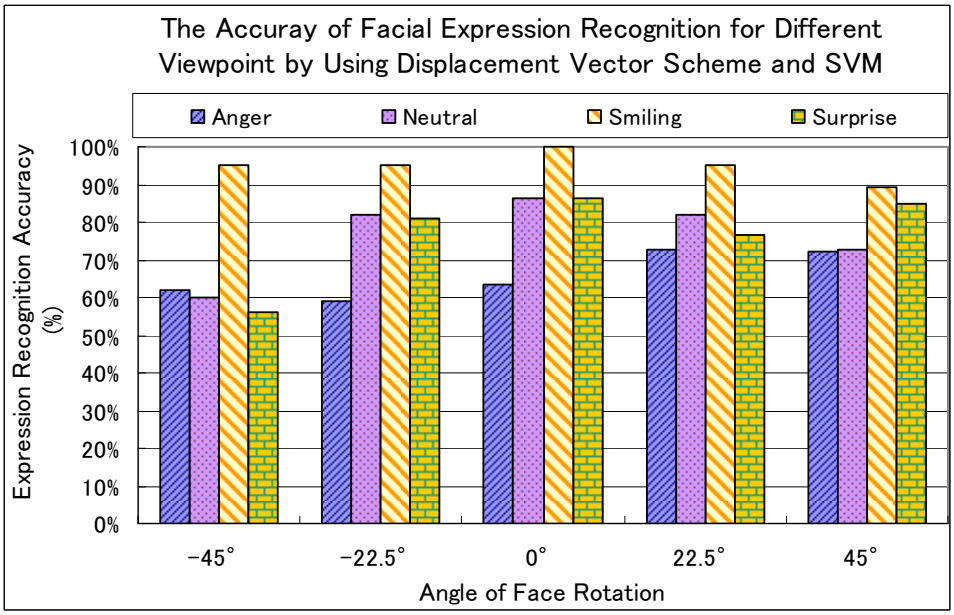

Figure 45. The accuracy of facial expression recognition for different face direction of -45°, -22.5°, 0°, +22.5° and +45° by using displacement vector scheme and SVM.

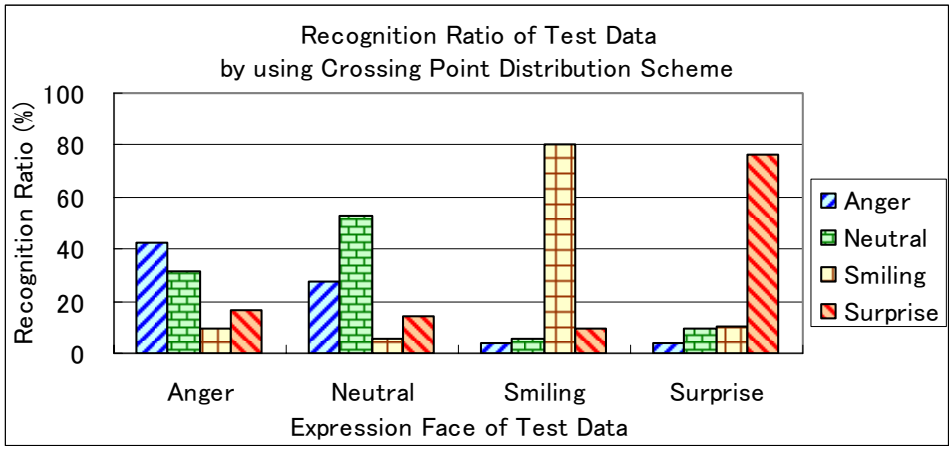

Figure 46. The recognition ratio of each expression face by using crossing point distribution scheme

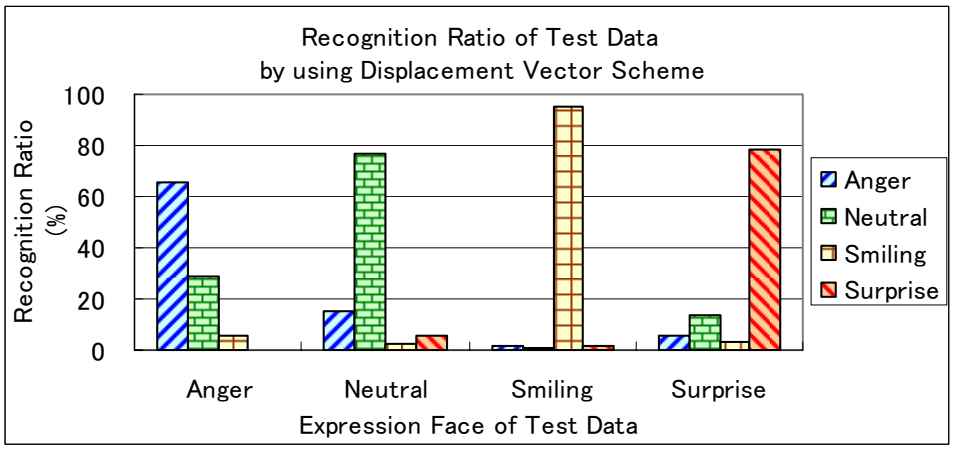

Figure 47. The recognition ratio of each expression face by using displacement vector scheme

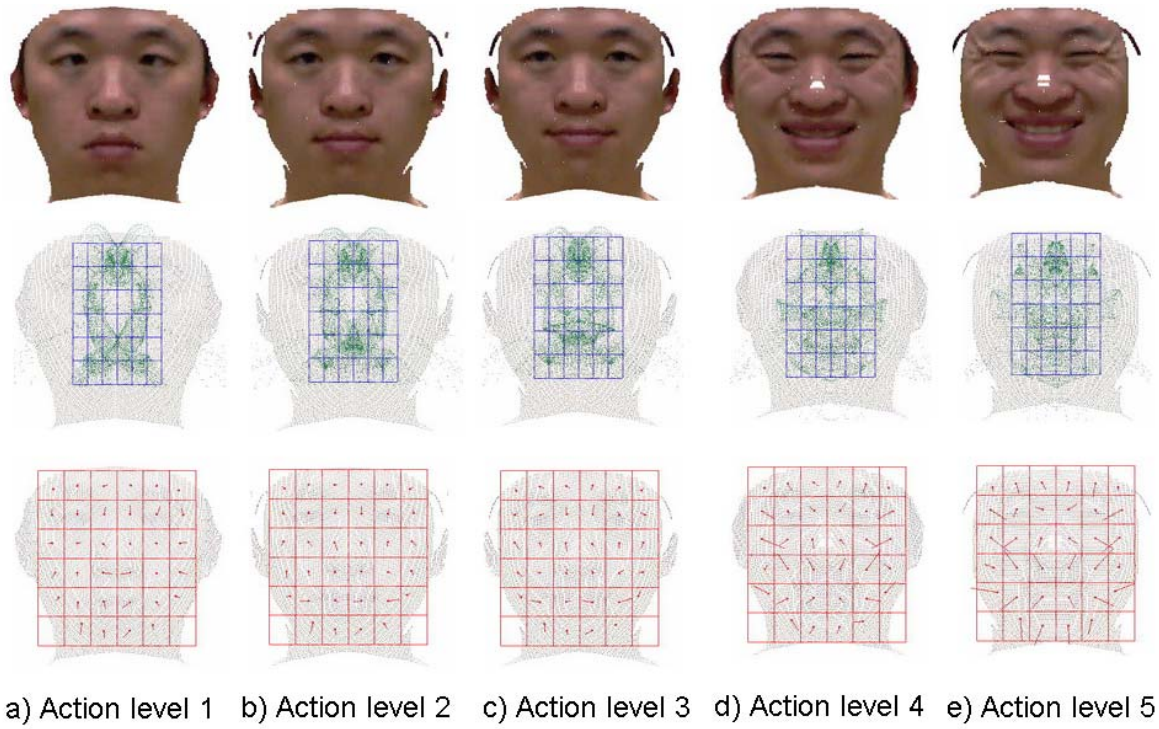

Figure 48. Characteristics of the crossing point distribution and displacement vector of a smiling face from a person in five action levels starting from neutral in a) to biggest smiling in e). The first row is the expression face. The second row is the distributions of the crossing point. The third row is the characteristics of the displacement vector.

4.3.2. Comparison of Viewpoint

The results in Table 3 show the highest accuracy at 0° degree and tend to be slightly low at greater angles. The average recognition rates were 51% and 69% at -45° and 63% and 80% at +45° by using the crossing point distribution method and displacement vector method, respectively.

We mentioned that only the successful data of 3D reconstruction were used for facial expression recognition in 4.3.1. However, each successful data has a slight reconstruction error. To evaluate the facial reconstruction error, a virtual face was aligned to the frontal real face according to six degrees of freedom (x, y, z, roll, pitch and yaw). The nose tip is used to rigidly align two face positions. The nose ridge vector (V_{nr}) is used to align the two faces to the yz-plane for a degree of freedom in roll. Then the virtual face is rotated around the y- and x-axes (yaw and pitch alignment) as being the origin point at the nose tip to

minimize the average matching distance. The virtual face and the frontal real face have been matched.

The average matching distances for each viewpoint of the virtual faces are shown in Fig. 37. The average matching distances of greater angles tend to be bigger than the one of frontal face. While the recognition accuracy also tend to be lower for the face captured from -45° and +45°. This tendency occurs because the reconstructed faces of the greater angers are fatter than the normal.

4.4 Discussion

A) Failed Nose Ridge Detection. The reconstruction of some samples without the cap failed as shown in Table 1. This failure occurs because in some cross sections of the face, some of the data is lost because of the black hair. Consequently, the semi-major axis vector on these cross sections points away from the nose direction. The data derived from such situations is not enough to give the correct shape of the ellipse and the correct direction of the semi-major axis vector. In this case, ten samples failed due to the lack of sufficient data.

B) Matching Distance. As the average matching distances of the virtual faces from different viewpoints are indicated in Fig.37, the data from greater angles tend to have bigger average matching distances because it is difficult to precisely estimate the rotation angle $\theta/2$ between the mirror face and the real face when comparing the smaller overlapped areas (see Fig. 16).

C) Head Pose in Roll and Pitch. The data from 140 faces posed in different rolls and pitches were tested. 11 samples indicated false nose ridge detection. The other faces were reconstructed successfully. Some results of the facial data were captured from -25 degrees (facing down), +25 degrees (facing up), -10 degrees (tilting right) and +10 degrees (tilting left) as shown in Fig. 49. These results show the detection of the nose ridge and of the symmetry plane on the face. They show the feasibility of this method when the head pose varies in roll and pitch.

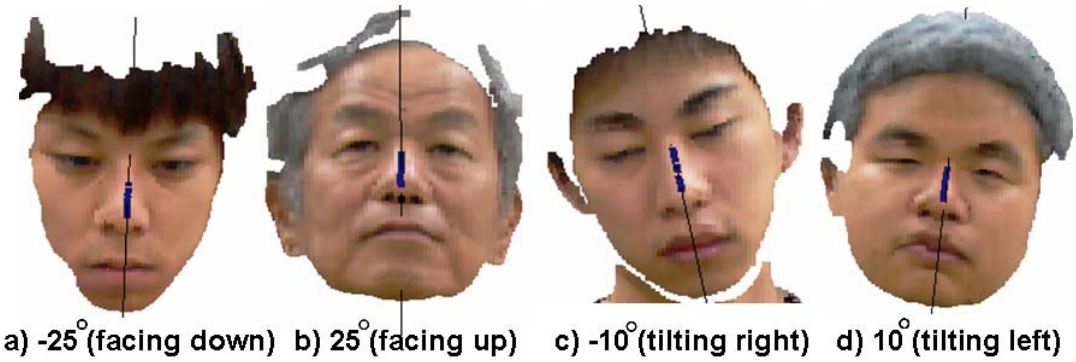

Figure. 49. Results of the nose ridge detection and symmetry plane correction of the real face in different head poses. The blue points indicate the detected nose ridge points. The line through the nose ridge points is the symmetry plane. The face is captured from -25 degrees (facing down), +25 degrees (facing up), -10 degrees (tilting right) and +10 degrees (tilting left).

D) Comparison of Two Facial Expression Recognition Methods. In this consideration, the performance of two facial expression recognition methods, crossing point distribution scheme and displacement vector scheme, is discussed.

The recognition accuracy of the displacement vector scheme and the crossing point distribution scheme is compared as shown in Figs. 44, 45, 50 and 51. Fig. 50 shows the accuracy for each viewpoint. Fig. 51 shows the accuracy for each expression face. The accuracy of the displacement vector scheme is higher because the displacement vector can represent the direction change of facial expression change. However, the computation time of the displacement vector scheme is longer than the one of crossing point distribution scheme. As the displacement vector is computed from the two points on the surface of the expression face and the neutral face which have the same xy-coordinate, the expression face has to be aligned to the neutral face. Therefore the longer computation time is spent for aligning the expression face.

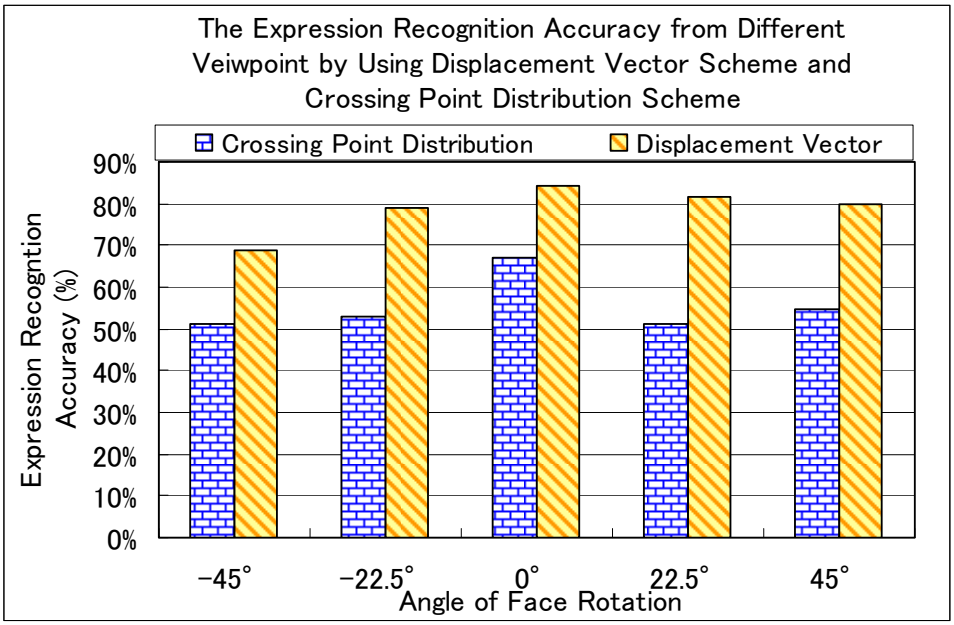

Figure 50. The expression recognition accuracy by displacement vector scheme and crossing point distribution scheme for different angle of face rotation

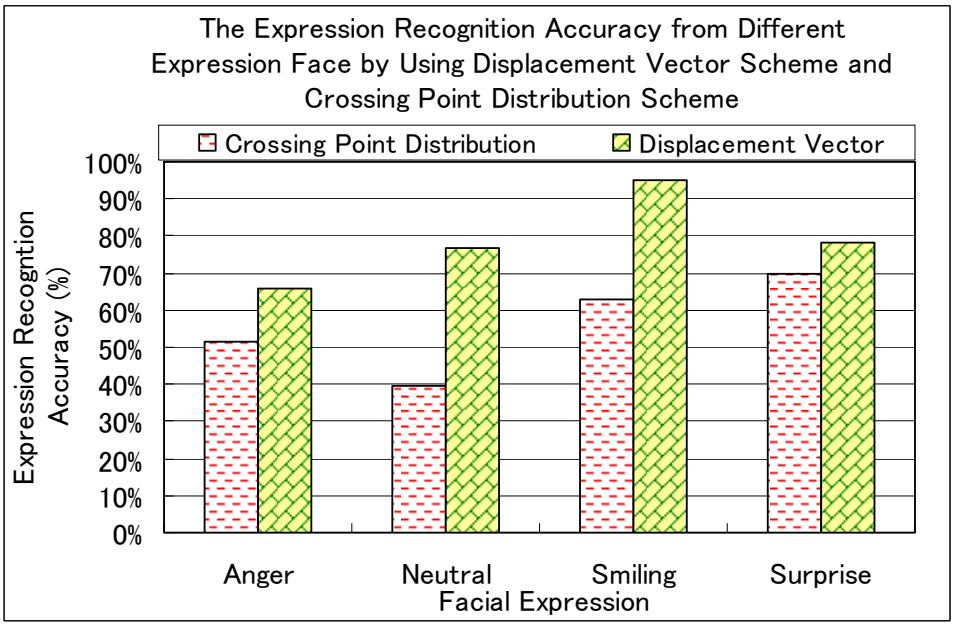

Figure 51. The expression recognition accuracy by displacement vector scheme and crossing point distribution scheme for different expression face

Chapter 5 Conclusion

In this paper, a novel method for facial expression recognition from a 2.5D partial face data set was proposed. The proposed method consists of three important algorithms: 1) nose ridge detection, 2) 3D face reconstruction and 3) facial expression recognition.

In proposed method, the nose ridge detection algorithm is applied to facial data captured from any viewpoint ranging from -80 degrees to +80 degrees in yaw. This algorithm was applied to the facial data from two situations one in which the hair is covered by a cap and another where it is not covered by a cap. The accuracy of nose tip detection was smaller than 1.0 mm. The inclination of the symmetry plane from the ideal symmetry plane was less than 2.1 degrees. The nose ridge detection algorithm was effective in detecting the nose tip and the nose ridge even when only half of the data from the face can be obtained.

For the 3D face reconstruction, this algorithm was also applied to facial data captured from any viewpoint ranging from -45 degrees to +45 degrees in yaw with a cap and without a cap. However, some of the data without a cap showed a performance slightly low due to the influence of messy hair. Therefore, for the moment, proposed method is best applied to people with short hair or with a cap. All of the matching distance results from the reconstructed faces that were correct were less than 2.0 mm. In the worst results in which the faces were too fat, the longest matching distance was 2.9 mm which is comparable to the results of Ref. 21).

In our facial expression recognition, the successful 3D face reconstruction data are used for the facial expression recognition. The facial expression is represented in terms of the change of crossing points on the face plane. Two crossing point analysis schemes, which are a crossing point distribution scheme and a displacement vector scheme, are used to analyze the crossing point for recognition. The experiments were done for four facial expressions (neutral, anger, surprise and smiling) from 22 persons based on the assumption that the person is known. The average recognition accuracy using the crossing point distribution scheme and the displacement vector scheme were 62.6% and 78.9% respectively. The advantages of the proposed method is that only one 2.5D partial face data

set captured from any viewpoint of face rotation between -45 degrees and + 45 degrees can be used for the facial expression recognition.

The proposed method can be applied to many applications including: 1) completely filling in a 3D facial data set that was lacking or missing some data[7], 2) facial expression recognition from a partial face data set to be used for robot perception and 3) facial recognition when only a side view of the face is available to be used with robot vision and human-machine interfaces.

References

[1] M. Pantic, and L.J.M. Rothkrantz, "Automatic Analysis of Facial Expressions: the State of the Art", *IEEE Trans. Pattern Analysis and Machine Intelligence*, Vol. 22, No. 12, pp. 1424-1445, Dec. 2000.

[2] B. Fasel, and J. Luettin, "Automatic Facial Expression Analysis: A Survey", *Pattern Recognition*, 36(1):259-275, 2003.

[3] X. Lu, A.K. Jain and D. Colbry, "Matching 2.5D Face Scans to 3D Models", *IEEE Trans. on Pattern Anal. Mach. Intell.*; 28(1), pp. 31-43 (2006).

[4] D.O. Gorodnichy, "On Importance of Nose for Face Tracking", *Proc. of IEEE Int. Conf. of Automatic Face and Gesture Recognition (FGR'02)*, pp. 181-186, Washington DC (2002).

[5] S. Gurbuz, K. Kinoshita and S. Kawato, "Real-time Human Nose Bridge Tracking in Presence of Geometry and Illumination Changes", *Proc. of IWMMS2004: The 2nd Int. Workshop on Man-Machine Symbiotic System*, (2004).

[6] C. Hesher, A. Srivastava, and G. Erlebacher, "Principal Component Analysis of Range Images for Facial Recognition", *Proc. of Int.'l Conf. on Imaging Science, Systems, and Technology (CISST)*, Nevada (2002).

[7] S. Malassiotis, and M.G. Strintzis, "Robust Real-Time 3D Head Pose Estimation from Range Data", *Pattern Recognition.;* 38 (8), pp. 1153-1165 (2005).

[8] C. Xu, T. Tan, Y. Wang, and L. Quan, "Combining Local Features for Robust Nose Location in 3D Facial Data", *Pattern Recognition Letter*, In Press Corrected Proof, 2006.

[9] D. Onofaio, S. Tubaro, A. Rama and F. Tarres, "3D Face Reconstruction with a four camera acquisition system", *Proc. Int. Workshop on Very Low Bitrate Video Coding (VLBV05)*, Costa Rei, Sardinia, 2005.

[10] T. Akimoto, Y. Suenaga, and R.S. Wallace, "Automatic Creation of 3D Facial Models", IEEE Computer Graphics and Applications, Vol. 13, September 1993, pp: 16-22.

[11] J. Yan, H. Zhang, "Realistic Virtual Face and Body Synthesis", IAPR Workshop on Machine Vision Applications, 2000, Tokyo, Japan.

[12] G. Pujitha, Sh. Hisanari and S. Yukio, “3-D Face Modeling with Multiple Range Image”, 15th Int. Conf. on Pattern Recognition (ICPR’00), Vol. 4, 2000, p. 4599.

[13] Y. Ben and S. J. Jesse, “A Novel Approach of 3D Reconstruction of Human Face Using Monocular Camera”, Proc. 11th Int. Multimedia Modelling Conference (MMM’05), 2005, p. 378-385.

[14] L. Yuan and L. G. Marina, “3D Facial Model Synthesis using Voronoi Approach”, 3rd Int. Symposium on Voronoi Diagrams in Science and Engineering (ISVD’06), 2006, pp: 132-137.

[15] Z. Liu, Z. Zhang, C. Jacobs and M. Cohen, “Rapid Modeling of Animated Faces from Video”, Technical Report MSR-TR-2000-11, http://research.microsoft.com/~zhang, 2000.

[16] P. Fua and C. Miccio, “From Regular Images to Animated Heads: A Least Squares Approach”, In Eurographics of Computer Vision, 1996, pp: 188-202.

[17] X. Lu and A.K. Jain, “Integrating Range and Texture Information for 3D Face Recognition”, *Proc. of the 7th IEEE Workshop on Applications of Computer Vision (WACV/MOTION’05)*, Vol. 1, pp. 156-163, USA (2005).

[18] K. Yamauchi and Y. Sato, “3D Human Body Measurement by Multiple Range Images”, *18th Int. Conf. on Pattern Recognition (ICPR’06)*, Vol. 4, pp. 833-836, USA (2006).

[19] V. Blanz and T. Vetter, “A Morphable model for the Synthesis of 3D Faces”, SIGGRAPH, 1999, pp: 187-194.

[20] V. Blanz, S. romdhani, and T. Vetter, “Face Identification across Different Poses and Illuminations with a 3D Morphable Model”, In AFGR 2002, 2002, pp. 192-197.

[21] V. Blanz, T. Vetter, “Reconstructing the Complete 3D Shape of Faces from Partial Information”, *Graphics Technical Report No. 1*, University of Freiburg, 2001.

[22] Zicheng Liu, “A Fully Automatic System to Model Faces from a Single Image”, Technical Report MSR-TR-2003-55.

[23] Dalong Jiang, Yuziao Hu, Shuicheng Yan, Lei Zhang, Hongjiang Zhang and Wen Gao, “Efficient 3D Reconstruction for Face Recognition”, Pattern Recognition, Vol. 38, No. 6, Jan. 2005, pp: 787-798.

[24] R. Zhang, P. Tsai, J. Cryer and M. Shah, “Shape from Shading: a Survey”, IEEE Trans. PAMI, 21(8): 690-706, 2002.

[25] Yi Liao and Rongchum Zhao, “Analysis and Evaluation of Several Typical SFS algorithms”, Journal of Image and Graphics, 6(10), 2001, pp: 953-961.

[26] P. Tsai and M. Shah, “Shape from Shading using linear approximation”, Journal of Image and Vision Computing, 12(8), 1994, pp. 487-498.

[27] H. Yuankui, Zh. Ying, and W. Zengfu, “Reconstruction of 3D Face from a Single 2D Image for Face Recognition”, Proc. 2nd Joint IEEE International Workshop on VS-PETS, Beijing, October 15-16, 2005.

[28] P. Besl and N. McKay, “A Method for Registration of 3D Shapes”, *IEEE Trans. Pattern Anal. Mach. Intell.*; 14(2), pp. 239-256 (1992).

[29] C. Dorai, G. Wang, A.K. Jain, and C. Mercer, “Registration and Integration of Multiple Object Views for 3D Model Construction”, *IEEE Trans. Pattern Anal. Mach. Intell.*; 20 (1), pp. 83-89 (1998).

[30] G. Turk and M. Leyoy, “Zippered Polygon Meshes from Range Images”, *Proc. of SIGGRAPH’94*, pp. 311-318, USA (1994).

[31] M. Pantic, and L.J.M. Rothkrantz, “Expert System for Automatic Analysis of Facial Expression”, *Image and Vision Computing Journal*, 18(11):881-995, 2000.

[32] H. Micheal, A. Wiratanaya and K.F. Kraiss, “Facial Expression Modeling from Still Images using a Single Generic 3D Head Model”, Proceeding of the 28th DAGM Symposium, Berling, pp. 324-333, September, 2006.

[33] I.A. Essa, A.P. Pentland, “Coding, Analysis, Interpretation, and Recognition of Facial Expressions”, IEEE Trans. Pattern Analysis and Machine Intelligence, Vol. 19, No. 7. pp. 757-763, July, 1997.

[34] Li and A. Barreto, “An Integrated 3D Face-Expression Recognition Approach”, Proceeding of ICASSP2006, Toulouse, 2006.

[35] J. Wang, L. Yin, X. Wei and Y. Sun, “3D Facial Expression Recognition Based on Primitive Surface Feature Distribution”, Proceeding of CVPR’06, New York, pp. 1399-1406, October, 2006.

[36] R. Halir, and J. Flusser, “Numerically Stable Direct Least Squares Fitting of Ellipses”,

In Skala, V (ed) Proc. of Int. Conf. in Central Europe on Computer Graphics, Visualization and Interactive Digital Media, pp. 125-132 (1998).

[37] M. Benz, X. Laboureux, T. Maier, et al, "The Symmetry of Faces", *Proc. of VMV'2002, Erlangen, Germany,* pp. 43-50 (2002).

[38] H. Hase, M. Yoneda, T. Kasamatsu and J. Kato, "A Proposal for a Face Plane", *Proceeding of the 7th International Conference on Image Processing*, pp. 333-336, Greece, 2001.

[39] Tajima J., "Uniform color scale applications to computer graphics", *Computer Vision Graph Image Process*, Vol. 21, No. 3, Mar. 1983, pp. 305-325.

[40] C.-C. Chang and C.-J. Lin, "LIBSVM: a library for support vector machines". http://www.csie.ntu.edu.tw/~cjlin/libsvm.

Acknowledgments

I wish to express my sincerely gratitude to my academic advisor, Professor Hiroyuki Hase (Department of Fiber Amenity Engineering, University of Fukui) and Associate Professor Shogo Tokai (Department of Fiber Amenity Engineering, University of Fukui) for their continuous guidance and constructive suggestions throughout this study.

I would like to thank the students of the Department of System Design Engineering for the various ways they helped bring this work to fruition. I also appreciate the encouragement and financial support from Japanese Government for this study.

Finally, I wish to thank my family for their never-ending love and encouragement.

Publication Lists

Papers

1. **Th. Charoenpong**, T. Shogo and H. Hase, “3D Face Reconstruction by 2.5D Partial Face Data”, *The Journal of the IIEEJ*, Vol. 37, No. 4, 2008.

International Conferences

1. **Ch. Theekapun**, H. Hase and Sh. Tokai, “Robust Nose Localization by Using Fitting of Ellipse”, *Int. Conf. IEEE TENCON 2006*, Hong Kong, Nov 14-17, 2006.

2. **Ch. Theekapun**, H. Hase and Sh. Tokai, “A Novel Approach of 3D Face Reconstruction Using Ellipse Fitting”, *IAPR Conf. on Machine Vision Application (MVA2007)*, pp: 528-531, Tokyo, May, 2007.

3. **Ch. Theekapun**, Sh. Tokai and H. Hase, “Facial Expression Recognition from a Side View Face by Using Face Plane”, *Int. Conf. IEEE ICWAPR2007*, pp: 1096-1101, Beijing, Nov, 2007.

4. **Ch. Theekapun**, Sh. Tokai and H. Hase, “Facial Expression Recognition from a Partial Face Image by Using Displacement Vector”, *Int. Conf. ECTI-CON 2008*, pp: 441-444, Thailand, 2008.

National Conferences

1. **Ch. Theekapun**, H. Hase and Sh. Tokai, “3D Reconstruction from a 2.5D Facial Image”, 2006 Joint Conference of Hokuriku Chapters of Electrical Societies, Kanazawa, Japan, 2006.

2. **Ch. Theekapun**, H. Hase and Sh. Tokai, “Facial Symmetry Axis Detection by Fitting of Ellipse”, Conf. Pattern Recognition and Media Understanding (PRMU), Technical Report of IEICE, Fukui, Japan, March 13-15, 2006.

3. **Ch. Theekapun**, H. Hase and Sh. Tokai, “3D Face Reconstruction and Evaluation”, 2007 Joint Conference of Hokuriku Chapters of Electrical Societies, Fukui, Japan, 2007.

4. **Ch. Theekapun**, Sh. Tokai and H. Hase, “Novel Method of Facial Expression Recognition from a 2.5D Partial Face Image”, Conf. Pattern Recognition and Media Understanding (PRMU), pp.477-482, Kanazawa, Japan, January, 2008.

5. **Th. Charoenpong,** Sh. Tokai and H. Hase, “Facial Expression Recognition from a 2.5D Partial Face Image”, Meeting on Image Recognition and Understanding 2008 (MIRU2008), Nagano, Japan, July, 2008.